



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1976

Sampled data adaptive digital computer
control of surface ship maneuvers.

Uhrin, John Joseph

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/17932>

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

SAMPLED DATA ADAPTIVE DIGITAL COMPUTER
CONTROL OF SURFACE SHIP MANEUVERS

John Joseph Uhrin

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
PACIFIC GROVE, CALIF 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

SAMPLED DATA ADAPTIVE DIGITAL COMPUTER
CONTROL OF SURFACE SHIP MANEUVERS

by

John Joseph Uhrin III

June 1976

Thesis Advisor:

George J. Thaler

Approved for public release; distribution unlimited.

T174033



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Sampled Data Adaptive Digital Computer Control of Surface Ship Maneuvers		5. TYPE OF REPORT & PERIOD COVERED Engineer's Thesis; June 1976
7. AUTHOR(s) John Joseph Uhrin III		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 371
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Underway Replenishment, Ship Dynamics, Hydrodynamic Interaction, Destroyer Maneuvering, Simulation, Ship Propulsion, Hydrodynamics, Sea State, Low Order Models, Ship Speed Control, Optimal Control, Ship System Modeling, Replenishment At Sea, Nonlinear Control,		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The replenishment at sea (RAS) maneuver is studied in detail for heading and speed control. design of purposefully nonlinear control laws is accomplished for the Mariner hull using the linearized equations of motion in three degrees of freedom. Extensive use of low order modeling and optimal control theory was made. Procedure steps are presented in detail to facilitate redesign for other ship types. The results are verified using		



20.

DSL simulation for a number of possible RAS scenarios. The control systems are also tested in a sea state to insure proper operation in the presence of external perturbations.

19.

Automatic Control, Control System Design, Adaptive Control, Automated Ship Control, Digital Computer Control



Sampled Data Adaptive Digital Computer Control
of
Surface Ship Maneuvers

by

John Joseph Uhrin III
Lieutenant, United States Navy
B.E.E., Villanova University, 1967
M.S., Naval Postgraduate School, 1975

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1976

Thesis

U15

C 1

ABSTRACT

The replenishment at sea (RAS) maneuver is studied in detail for heading and speed control. Design of purposefully nonlinear control laws is accomplished for the Mariner hull using the linearized equations of motion in three degrees of freedom. Extensive use of low order modeling and optimal control theory was made. Procedure steps are presented in detail to facilitate redesign for other ship types. The results are verified using DSL simulation for a number of possible RAS scenarios. The control systems are also tested in a sea state to insure proper operation in the presence of external perturbations.



TABLE OF CONTENTS

FORM DD 1473.....	1
APPROVAL FORM.....	3
ABSTRACT.....	4
TABLE OF CONTENTS.....	5
LIST OF TABLES.....	7
LIST OF FIGURES.....	8
TABLE OF TERMS AND ABBREVIATIONS.....	14
ACKNOWLEDGEMENTS.....	16
I. INTRODUCTION.....	17
II. MODELING.....	19
A. MAFINDER DYNAMICS.....	19
B. RUDDER RESPONSE.....	32
C. ENGINE RESPONSE.....	38
D. EXTERNAL FORCES.....	43
1. Two Ships in Proximity.....	43
2. Waves.....	48
III. REFRESHMENT AT SEA.....	63
A. HEADING CONTROL.....	63
1. Control Choice.....	63
2. Control Method.....	64
3. Optimization.....	74
a. Technique.....	74
b. Cost Function.....	76
c. Results.....	77
d. Control Testing.....	85
e. Second Optimization.....	89
f. Continued Control Testing.....	97
g. Varying Initial Conditions.....	112
h. Performance in Sea State.....	138
B. VELOCITY CONTROL.....	156
1. Type of Control.....	157
2. Optimization.....	159



TABLE OF CONTENTS (cont.)

3.	Control Testing.....	163
4.	Longitudinal Position Offset.....	173
5.	Wave Effects on Velocity Control.....	194
IV.	CONCLUSIONS AND RECOMMENDATIONS.....	199
	A. CONCLUSIONS.....	199
	E. RECOMMENDATIONS.....	201
APPENDIX A.....		203
APPENDIX E.....		289
APPENDIX C.....		298
COMPUTER PROGRAM #1.....		320
COMPUTER PROGRAM #2.....		324
COMPUTER PROGRAM #3.....		327
COMPUTER PROGRAM #4.....		329
COMPUTER PROGRAM #5.....		332
COMPUTER PROGRAM #6.....		338
COMPUTER PROGRAM #7.....		344
COMPUTER PROGRAM #8.....		350
COMPUTER PROGRAM #9.....		356
COMPUTER PROGRAM #10.....		362
BIBLIOGRAPHY.....		368
INITIAL DISTRIBUTION.....		370



LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
II-1	Symbols and Nomenclature	25
II-2	Characteristics of Mariner-type Ship	27
II-3	Non-dimensional Hydrodynamic Coefficients	28
II-4	Rudder Command and Response	35
II-5	Wave Simulation Listing	56
III-1	Approach Phase Optimization Results	78
III-2	Turn Phase Optimization Results	90
III-3	Initial Condition Simulation	113
	Cross Reference	
III-4	Optimization Results	161
III-5	Polynomial Curve Fit Results	164
III-6	Position Offset Testing	174
	Cross Reference	
C-1	Interactive Curve Fit Polynomial Coefficients	299
C-2	Interactive Curve Fit Error Analysis	303

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
II-1	Direction and Sense of Hydrodynamic Terms	21
II-2	Rudder Step and Ramp Model Geographic Comparison	30
II-3	Rudder Step and Ramp Model Yaw Difference vs. Time	31
II-4	Rudder Block Diagram	33
II-5	Rudder Responses	36
II-6	Rudder Responses	37
II-7	Gas Turbine Propulsion Plant Block Diagram	39
II-8	Propulsion Plant High Order Model Step Response	40
II-9	Propulsion Plant Low Order Model Block Diagram	38
II-10	Step Response Comparison of Low and High Order Propulsion Plant Models	42
II-11	Family of Interactive Y Force Curves	45
II-12	Family of Interactive N Moment Curves	46
II-13	Interactive Forces Effect on the Geographic Plot	52
II-14	Lateral (Y) Forces	53
II-15	Rotational (N) Moments	54
II-16	Interactive Forces Effect on Yaw of the Control Ship	55
II-17	Wave Simulation Run #1	57
II-18	Wave Simulation Run #2	58
II-19	Wave Simulation Run #3	59
II-20	Wave Simulation Run #4	60



LIST OF FIGURES (cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
II-21	Wave Simulation Run #5	61
II-22	Wave Simulation Run #6	62
III-1	Measurement Technique	65
III-2	Alternate Method of Measurement	67
III-3	Distance Logic	69
III-4	Heading Difference Calculation	70
III-5	Rudder Control Block Diagram	73
III-6	Optimization Flow Chart	75
III-7	Approach Phase Yaw Result	79
III-8	Approach Phase Y Forces	80
III-9	Approach Phase N Moments	81
III-10	Approach Phase Geographic Plot	82
III-11	Approach Phase Lateral Distance DY	83
III-12	Approach Phase Rudder Response	84
III-13	Turn Phase Rudder Action of Reference Ship	86
III-14	Turn Phase Rudder Response	87
III-15	Turn Phase Lateral Distance DY	88
III-16	Turn Phase Yaw Response	91
III-17	Turn Phase Y Forces	92
III-18	Turn Phase N Moments	93
III-19	Turn Phase Geographic Plot	94
III-20	Turn Phase Lateral Distance DY	95
III-21	Turn Phase Rudder Response	96
III-22	Approach Phase Lateral Distance DY	98
III-23	Approach Phase Yaw Response	100
III-24	Approach Phase Y Forces	101
III-25	Approach Phase N Moments	102
III-26	Approach Phase Geographic Plot	103
III-27	Approach Phase Lateral Distance DY	104
III-28	Approach Phase Rudder Response	105



LIST OF FIGURES (cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
III-29	Turn Phase Yaw Response	106
III-30	Turn Phase Y Forces	107
III-31	Turn Phase N Moments	108
III-32	Turn Phase Geographic Plot	109
III-33	Turn Phase Lateral Distance DY	110
III-34	Turn Phase Rudder Response	111
III-35	Approach Phase Run #1 Yaw Response	114
III-36	Approach Phase Run #1 Geographic Plot	115
III-37	Approach Phase Run #1 Rudder Response	116
III-38	Approach Phase Run #2 Yaw Response	117
III-39	Approach Phase Run #2 Geographic Plot	118
III-40	Approach Phase Run #2 Rudder Response	119
III-41	Approach Phase Run #3 Yaw Response	120
III-42	Approach Phase Run #3 Geographic Plot	121
III-43	Approach Phase Run #3 Rudder Response	122
III-44	Approach Phase Run #4 Yaw Response	123
III-45	Approach Phase Run #4 Geographic Plot	124
III-46	Approach Phase Run #4 Rudder Response	125
III-47	Approach Phase Run #5 Yaw Response	126
III-48	Approach Phase Run #5 Geographic Plot	127
III-49	Approach Phase Run #5 Rudder Response	128
III-50	Approach Phase Run #6 Yaw Response	129
III-51	Approach Phase Run #6 Geographic Plot	130
III-52	Approach Phase Run #6 Rudder Response	131
III-53	Turn Phase Run #4 Yaw Response	132
III-54	Turn Phase Run #4 Geographic Plot	133
III-55	Turn Phase Run #4 Rudder Response	134
III-56	Turn Phase Run #6 Yaw Response	135
III-57	Turn Phase Run #6 Geographic Plot	136
III-58	Turn Phase Run #6 Rudder Response	137
III-59	Approach Phase Run #4 Lateral Y Forces	139
III-60	Approach Phase Run #4 Rotational N Moments	140



LIST OF FIGURES (cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
III-61	Approach Phase Run #4 Lateral Distance DY	141
III-62	Turn Phase Run #4 Lateral Y Forces	142
III-63	Turn Phase Run #4 Rotational N Moments	143
III-64	Turn Phase Run #4 Lateral Distance DY	144
III-65	Wave Effect on Turn Phase Lateral Distance (DY) WL=1.5	147
III-66	Wave Effect on Approach Phase Yaw WL=1.0	148
III-67	Approach Phase Rudder Response to Waves WL=1.0	149
III-68	Wave Effect on Approach Phase Lateral Distance (DY) WL=1.0	150
III-69	Approach Phase Wave Profile WL=1.0	151
III-70	Wave Effect on Turn Phase Yaw WL=1.0	152
III-71	Wave Effect on Turn Phase Lateral Distance (DY) WL=1.0	153
III-72	Turn Phase Rudder Response to Waves WL=1.0	154
III-73	Turn Phase Wave Profile WL=1.0	155
III-74	Ncn-optimum Speed Law	156
III-75	Speed Control Law	158
III-76	Optimization Flow Chart	159
III-77	Switching Curve Minimization Results	162
III-78	First thru Fifth Order Polynomial Curve Fit Results	165
III-79	Fifth Order Polynomial Curve Fit	166
III-80	RAS Speed Control Approach Phase Speed Desired (1) and Speed Acquired (2) vs. Real Time	168
III-81	RAS Speed Control Approach Phase Position Attainment vs. Real Time	169

LIST OF FIGURES (cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
III-82	RAS Speed Control Turn Phase Speed Desired (1) and Speed Acquired (2) vs. Real Time	170
III-83	RAS Speed Control Turn Phase Position Attainment vs. Real Time	171
III-84	Approach Phase Run A Lateral Distance DY	176
III-85	Approach Phase Run A Yaw Difference	177
III-86	Approach Phase Run B Lateral Distance DY	178
III-87	Approach Phase Run B Yaw Difference	179
III-88	Approach Phase Run B Speed Response	180
III-89	Approach Phase Run B Longitudinal Position DX	181
III-90	Approach Phase Run C Lateral Distance DY	182
III-91	Approach Phase Run C Yaw Difference	183
III-92	Approach Phase Run C Speed response	184
III-93	Approach Phase Run C Longitudinal Position DX	185
III-94	Turn Phase Run A Lateral Distance DY	186
III-95	Turn Phase Run A Yaw Difference	187
III-96	Turn Phase Run B Lateral Distance DY	188
III-97	Turn Phase Run B Yaw Difference	189
III-98	Turn Phase Run B Longitudinal Position DX	190
III-99	Turn Phase Run C Lateral Distance DY	191
III-100	Turn Phase Run C Yaw Difference	192
III-101	Turn Phase Run C Longitudinal Position DX	193
III-102	Block Diagram of Wave Introduction in Speed Loop	194
III-103	Approach Phase Speed Response in Waves	196
III-104	Turn Phase Speed Response in Waves	197
III-105	Turn Phase Longitudinal Position EX in Waves	198
C-1	Curve Fitted Interactive Y Forces	305
C-2	Curve Fitted Interactive N Moments	306



LIST OF FIGURES (cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
C-3	Approach Phase Curve Fitted Y Forces	308
C-4	Approach Phase Curve Fitted N Moments	309
C-5	Approach Phase Geographical Plot From Modified Interactive Effects	310
C-6	Turn Phase Curve Fitted Y Forces	312
C-7	Turn Phase Curve Fitted N Moments	313
C-8	Turn Phase Lateral Distance DY From Modified Interactive Effects	314



TABLE OF TERMS AND ABBREVIATIONS

ADX - DX referenced to control ship's head
ADY - DY referenced to control ship's head
Alongside - position at which longitudinal position ADX is
0.0
Approach Phase - phase in RAS scenario at which the control
ship comes alongside the reference ship
Approach Speed - speed at which the control ship will
commence approach to come alongside the
reference ship
AT - real time as referenced to the full size Mariner hull
Control Ship - ship making the RAS approach
Desired Distance - lateral distance at which RAS desired
DSL - Digital Simulation Language (IBM developed)
DX - center of ship's geographic separation along X axis
DY - center of ship's geographic separation along Y axis
Geographic Coordinates - earth's coordinate system
JCL - Job Control Language for IBM 360/67 computer
Kt., kts. - knot, knots - 1 nautical mile/hour or 2000
yards/hour
L, Ship length - length of one mariner hull used in this
thesis (527.8 feet)
Lateral Distance - equal to ADY
Longitudinal Distance/Position - equal to ADX
LUC - nondimensionalizing scaling factor
Port Side To - approach (control) ship replenishes with its
port side toward supply (reference) ship
RAS - Replenishment At Sea
Receiving Ship - control ship or ship B
Reference Ship - ship that maintains course and speed
Reference Speed - speed of reference ship

TABLE OF TERMS AND ABBREVIATIONS (cont.)

Replenishment Speed - signaled intended speed at which RAS
will be conducted

Stbd Side Tc - approach (control) ship replenishes with its
stbd side toward supply (reference) ship

Supply Ship - reference ship or ship B

T - nondimensionalized time used in DSL runs

Turn Phase - phase in RAS scenario at which the ships are in
their desired positions and the
reference ship is turned

Yaw - ship's heading in relation to true north

Y Coordinate - geographic reference system E is +, W is -

X Coordinate - geographic reference system N is +, S is -

ACKNOWLEDGEMENTS

In the course of thesis research many people contribute to the final product. It is not possible to afford individual credit to all that have given of themselves for this particular research. Some, however, have rendered assistance that has proven invaluable.

Professor G. J. Thaler has been, without a doubt, the prime guiding force that has kept this study within a sound perspective.

Three members of the computer center staff, M. Anderson, Kris Butler, and Ed Donnellan have tolerated numerous intimidations with good humor which made many otherwise arduous hours bearable. Without their professional assistance and personal contributions to the computer center operations, this thesis would have fallen far short of its goal.

And finally, my dear wife Mary who has endured a husband that has spent endless hours married to the IBM 360 computer. Her undying support and valuable encouragement is held in the highest esteem.

I. INTRODUCTION

The advent of the digital computer as standard equipment on board virtually all modern Naval ships has opened the field of Digital Computer Control in almost all aspects of ship life. The computer has been a viable asset in fire control systems for years and has been used extensively for aids to ship maneuvering control in the form of NTDS (Naval Tactical Data System) readouts. The declining costs of general and special purpose computers has made their inception as a manpower replacement or augmentation a reality. Their high speed and accuracy can make them perform functions with much greater safety than previously attainable with time proven (and sometimes time weary) "seaman's eye."

This then is the basis for this thesis; a study of the types of maneuvers that can be handled more accurately and safely than presently being accomplished.

One area of study is the total Replenishment At Sea (RAS) problem including both course and speed control for the approach and alongside phases. This situation has always been one of extreme danger due to the collision potentials involved. However, other dangers are involved in the on deck evolutions when the ships are not kept at a fairly constant distance. Sudden violent maneuvers may cause the stress on the lines between ships to increase enough to cause the lines to part. The reality of this danger is readily apparent to anyone who has ever seen a Manila line or steel cable part or a kingpost shackle break or a kingpost suddenly bend under these extreme stresses. A system which will minimize these potential dangers is well

worth investigation.

Of course with a digital computer, the algorithm for RAS can be modified or replaced to enable its use as a maneuvering control device for other situations such as formation steaming or single ship navigation transit control.

II. MODELING

A. MARINER DYNAMICS

In the conception of this thesis, realistic models of modern destroyer hull configurations were sought. This search proved fruitless. The hydrodynamic coefficients for present day destroyers are not currently available. However, some naval and civilian research is presently being conducted to obtain these coefficients.

A complete set of these coefficients is necessary for any maneuvering control system design. A hull configuration which has been under continual study with well defined and verified hydrodynamic coefficients was chosen^[1]. This hull is commonly referred to as the "Mariner" hull.

The development of the equations of motion in six and three degrees of freedom have been well documented^[2]. The model used for this thesis is the equations of motion in three degrees of freedom linearized with second order and higher terms eliminated. These equations are characterized by dependency on small perturbations about a specific operating point. The maneuvers experienced in the following chapters do not entirely meet this criterion. The inadequacy and shortcomings of this model are of little consequence because relevant hydrodynamic coefficients are not available, and the methods presented can be applied to any ship type.

The development of the model is readily available to the interested reader ^[3]; only a summary of the equations

and their corresponding hydrodynamic coefficients are presented here.

The equations of motion used are as follows:

$$(X_u - m) \ddot{U} + X_u (U - u_0) + X_d \dot{d} = 0$$

$$(Y_v - m) \ddot{V} + Y_v V + (Y_r - m) R + Y_r \dot{R} + Y_d \dot{d} = 0$$

$$(N_z - I) \ddot{\theta} + N_z R + N_v \dot{V} + N_v V + N_d \dot{d} = 0$$

The direction and sense of the terms in the above equations are shown in figure II-1. Letting:

$$a_{11} = m - Y_v$$

$$b_{11} = -Y_v$$

$$c_{11} = 0$$

$$a_{21} = -Y_r$$

$$b_{21} = m - Y_r$$

$$c_{21} = 0$$

$$a_{12} = -N_v$$

$$b_{12} = -N_v$$

$$c_{12} = 0$$

$$a_{22} = I - N_z$$

$$b_{22} = -N_z$$

$$c_{22} = 0$$

$$a_{33} = m - X_u$$



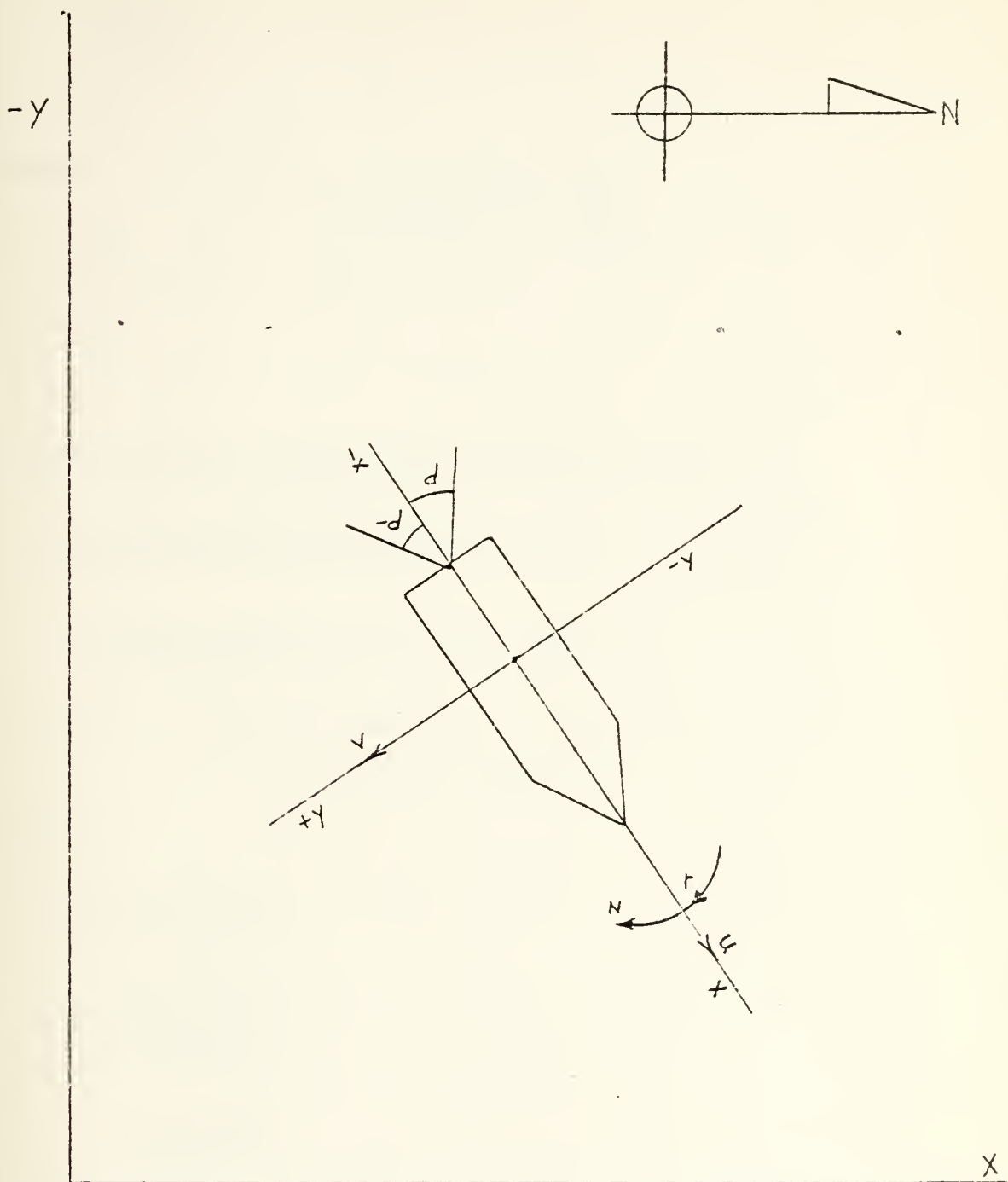


Figure II-1
Direction and Sense of Hydrodynamic Terms

$$b_{33} = -X_u$$

$$c_{33} = 0$$

Setting

$$V = \dot{A}$$

$$\Psi = B$$

$$U = \dot{C}$$

The equations of motion can be written as:

$$a_{11} \ddot{A} + b_{11} \dot{A} + c_{11} A + a_{21} \ddot{B} + b_{21} \dot{B} + c_{21} B = IF1$$

$$a_{12} \ddot{A} + b_{12} \dot{A} + c_{12} A + a_{22} \ddot{B} + b_{22} \dot{B} + c_{22} B = IF2$$

$$a_{33} \ddot{C} + b_{33} \dot{C} + c_{33} C = IF3$$

or:

$$a_{11} \ddot{A} + a_{21} \ddot{B} = I1$$

$$a_{12} \ddot{A} + a_{22} \ddot{B} = I2$$

$$a_{33} \ddot{C} = I3$$

where:

$$IF1 = -Y_d \cdot d$$

$$IF2 = N_d \cdot d$$



$$IF3 = - \int x_u dt$$

and:

$$I1 = -t_{11} \dot{A} - c_{11} A - b_{21} \dot{B} - c_{21} B + IF1$$

$$I2 = -t_{12} \dot{A} - c_{12} A - b_{22} \dot{B} - c_{22} B + IF2$$

$$I3 = -t_{33} \dot{C} - c_{33} C + IF3$$

By solving this system of equations, the following relationships are established:

$$A = (a_{22} a_{33} I1 - a_{33} a_{21} I2) / \text{DEL}$$

$$B = (a_{11} a_{33} I2 - a_{33} a_{12} I1) / \text{DEL}$$

$$C = I3 / a_{33}$$

where:

$$\text{DEL} = a_{33} (a_{11} a_{22} - a_{12} a_{21})$$

which yields the solution:

$$V = \dot{A} = v_0 + \int_{t_0}^t A dt$$

$$\psi = B = \psi_0 + \int_{t_0}^t \dot{B} dt = \psi_0 + \int_{t_0}^t [\dot{B}(0) + \int_{t_0}^t \dot{B} dt] dt$$

$$U = \dot{C} = u_0 + \int_{t_0}^t C dt$$

The space coordinate system is defined as follows:



$$\dot{Y} = U \cdot \sin \psi + V \cdot \cos \psi$$

$$\dot{X} = U \cdot \cos \psi - V \cdot \sin \psi$$

where:

$$X = X_0 + \int_{t_0}^t \dot{X} \, dt$$

$$Y = Y_0 + \int_{t_0}^t \dot{Y} \, dt$$

Table II-1 summarizes the symbols and nomenclature used in the foregoing abbreviated solution of motion in three degrees of freedom. The applicable characteristics of the Mariner hull are presented in table II-2 with the corresponding nondimensionalized hydrodynamic coefficients and the DSL computer program variable names delineated in table II-3.

Computer Program #1 is the basic DSL program that was developed from these equations of motion. This program uses two ships to illustrate the turning characteristics of the Mariner hull for various rudder commands. Figure II-2 shows the difference between a step model and a ramp model rudder in a geographic plot. Figure II-3 shows the corresponding difference in yaw.



TABLE II-1
SYMBOLS AND NOMENCLATURE

<u>Symbol</u>	<u>Definition</u>
$X_{\dot{u}}$	derivative of longitudinal force with respect to longitudinal acceleration \dot{u}
U_u	derivative of longitudinal force with respect to longitudinal velocity U
$Y_{\dot{v}}$	derivative of lateral force with respect to transverse acceleration \dot{v}
Y_v	derivative of lateral force with respect to transverse velocity v
$Y_{\dot{r}}$	derivative of lateral force with respect to angular acceleration \dot{r}
Y_r	derivative of lateral force with respect to angular velocity R
Y_d	derivative of lateral force with respect to rudder angle d
$N_{\dot{v}}$	derivative of yaw moment with respect to transverse acceleration \dot{v}
N_v	derivative of yaw moment with respect to transverse velocity v

TABLE II-1 (cont.)

SYMBOLS AND NOMENCLATURE

<u>Symbol</u>	<u>Definition</u>
$N_{\dot{r}}$	derivative of yaw moment with respect to angular acceleration \dot{R}
N_r	derivative of yaw moment with respect to angular velocity R
N_d	derivative of yaw moment with respect to rudder angle d
\dot{R}	yaw angle acceleration
R	yaw angle velocity
u_0	initial velocity of origin of body axes relative to fluid
\dot{v}	transverse acceleration of ship axes relative to fluid
v	transverse velocity of origin of ship axes relative to fluid
X	hydrodynamic longitudinal force
Y	hydrodynamic lateral force
\dot{u}	longitudinal acceleration of ship axes relative to fluid



TABLE II-1 (cont.)
SYMBOLS AND NOMENCLATURE

<u>Symbol</u>	<u>Definition</u>
U	longitudinal velocity of ship axes relative to fluid
ψ	yaw angle
A_t	actual time
T	nondimensionalized time
x_g	longitudinal distance that the ship center of gravity is forward of the ships axes
u_1	longitudinal velocity of ship axes relative to fluid (operating point)

TABLE II-2
CHARACTERISTICS OF MARINER-TYPE STUDY SHIP

Length, ft.	527.8
Beam, ft	76.0
Draft, ft	29.75
Displacement, tons	16,800.
Block coefficient, C_b	0.6



TABLE II-3

NONDIMENSIONAL HYDRODYNAMIC COEFFICIENTS

<u>Coefficient</u>	<u>Computer Program</u> <u>Variable Name</u>	<u>Nondimensional</u> <u>Value</u>
$(X_{-m})_u$	MXUD	-0.0085
X_u	XU	-0.0012
Y_v	YV	-0.01243
$(Y_{-m})_v$	MYVD	-0.015
$(Y_{-mu})_{r1}$	MYR	-0.0051
$(Y_{-ux})_{rg}$	YRD	-0.00027
Y_d	YDEL R	+0.0027
N_v	NV	-0.00351
N_v	NVD	-0.000197
$(N_{-mxu})_{rg1}$	NR	-0.00227
$(N_{-I})_{rz}$	IZNRD	-0.00068
N_d	NDEL R	-0.00126
X_n	---	-0.0000462
Y_n	---	-0.0000052
N_n	---	+0.0000026
X_d	XDEL R	0.0

NOTE: $x_g = 0.0$

TABLE II-3 (cont.)

NONDIMENSIONAL HYDRODYNAMIC COEFFICIENTS

Values based on the following operating point:

$$u_1 = 1.0 \text{ (15 Kts)}$$

$$\dot{\psi} = 0.0$$

$$\dot{v} = 0.0$$



Figure II-2
Rudder Step and Ramp Model Geographic
Comparison

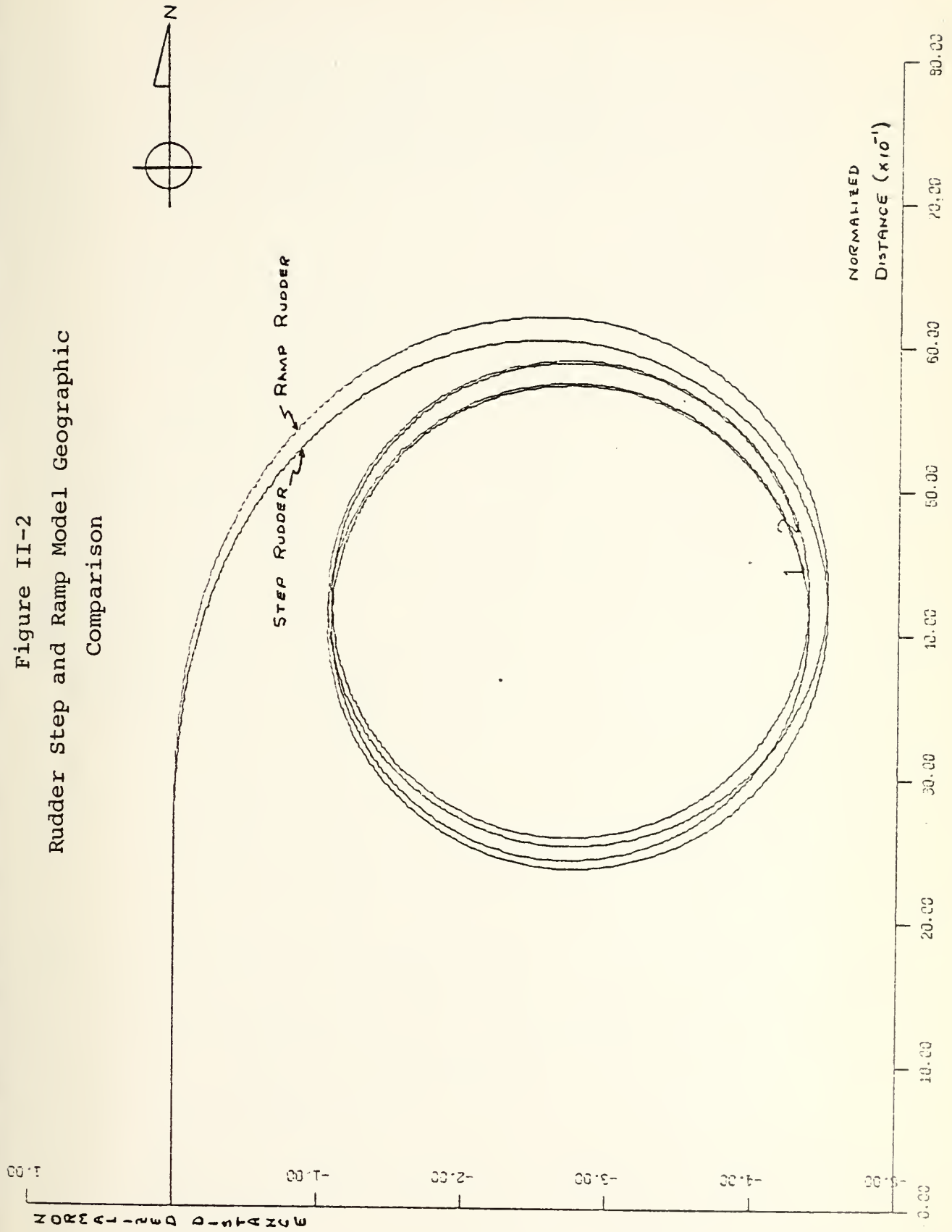
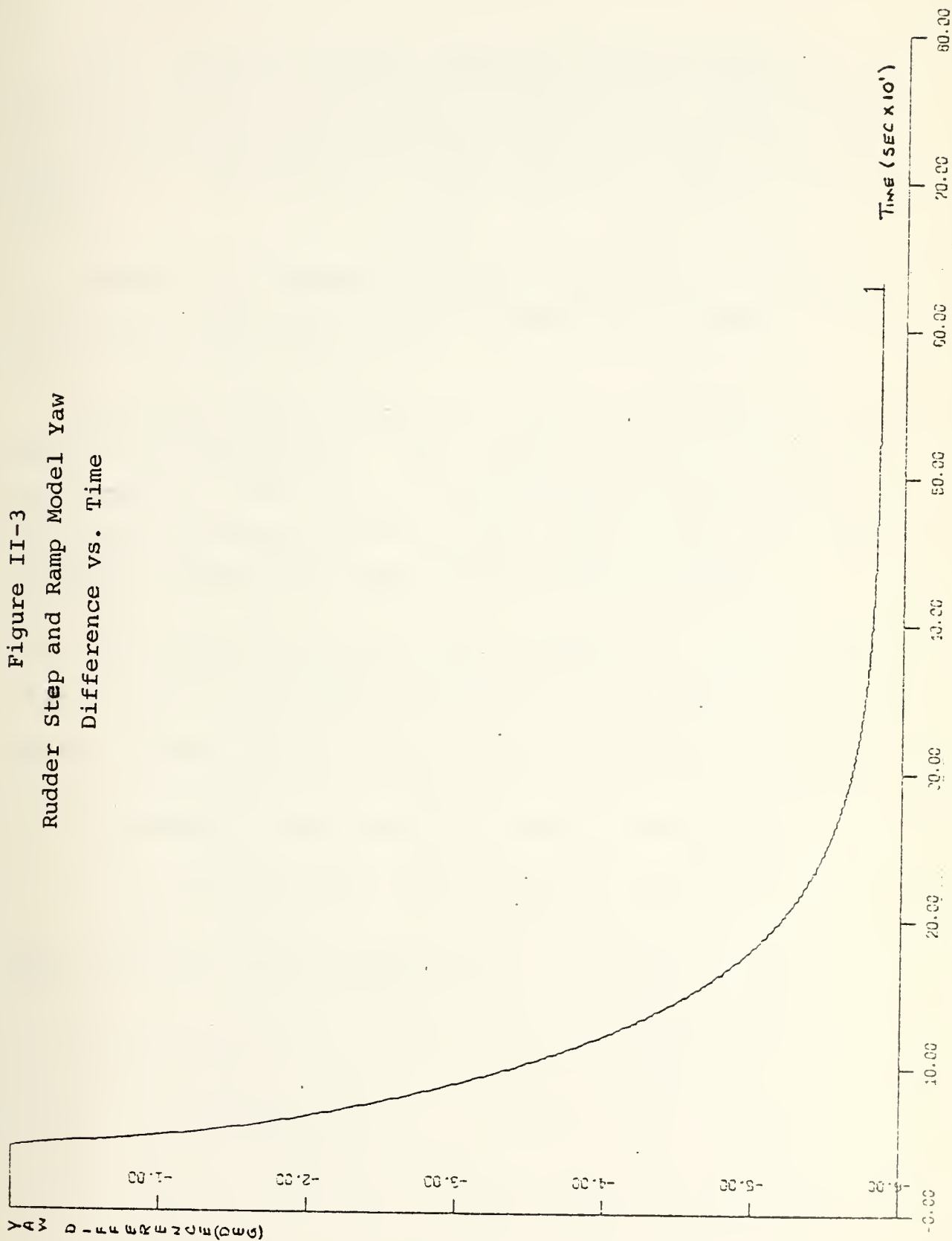




Figure II-3
Rudder Step and Ramp Model Yaw
Difference vs. Time



E. RUDDER RESPONSE

The previous section indicates a marked difference in behavior between step and ramp rudder models. This prompted an investigation into realistic rudder modeling which would fulfill the requirements of limit stops and maximum rudder rate.

NSREC[*] has modeled the rudder of the DD-931 Class Destroyer. The basics of this model are presented in the block diagram of figure II-4.

The first limiter models the rudder stops which for the Mariner are ± 30 degrees. The second limiter models the proportional band of a variable-displacement pump by limiting its maximum percent stroke. The limits for this nonlinear element have been found to be ± 7 degrees.

The transfer function (K_g/s) accepts an input error signal of up to 7 degrees, converts it to a rudder rate, and integrates the rudder rate to obtain rudder angle. Letting:

$$\dot{d}_m = \text{Maximum rudder rate (2.0 degrees/sec)}$$

$$d_{emax} = \text{Maximum error input (7.0 degrees)}$$

The system gain can be defined as:

$$\begin{aligned} K_g &= \dot{d}_m / d_{emax} \\ &= 2.0/7.0 \\ &= 0.285714 \text{ /sec} \end{aligned}$$



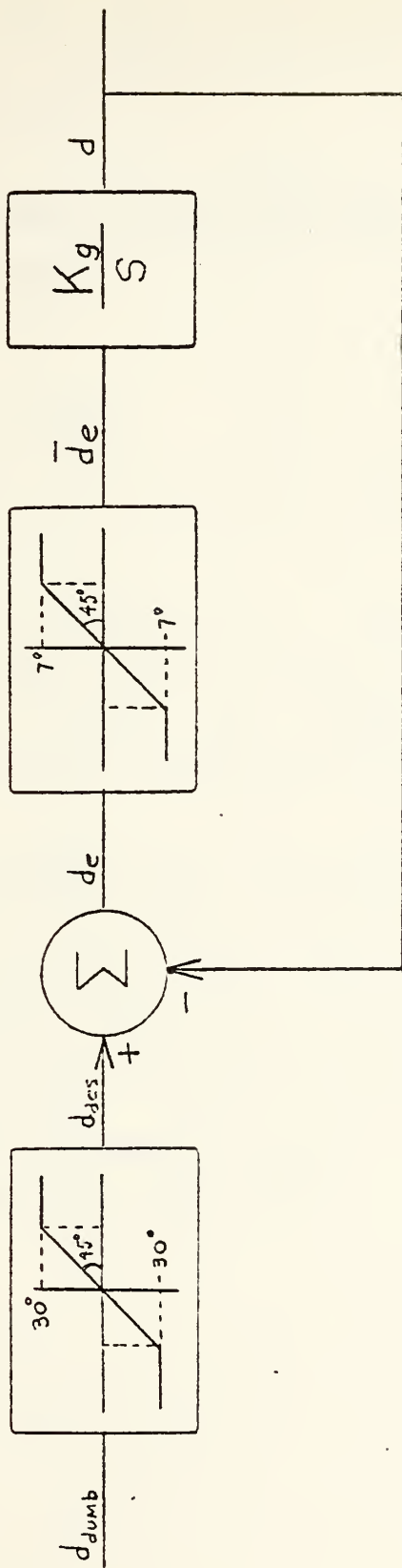


Figure II-4
Rudder Block Diagram

To convert this model to the required nondimensionalized form, the following manipulation is required:

$$\begin{aligned}K_g' &= K_g \cdot L/u_1 \\&= 0.285714 \cdot 527.8 / (15 \cdot 1.689) \\&= 5.95224\end{aligned}$$

where:

L = ship length

u_1 = operating point speed (15 Kts \cdot 1.689 ft/sec/Kt)

Computer Program #2 is the DSL program which models this system. The curves of figure II-5 and II-6 exhibit the responses of various step rudder commands. These are tabulated and cross referenced in table II-4.

These responses show the characteristics of a realistic rudder in that the rudder is never allowed to slam into the stops. They exhibit the time delay between command and response which is a function of the rate of response (2.0 degrees/sec). A control system design with this scheme is a much more difficult problem than one with an idealized rudder (step response) because the entire rudder control system becomes quite nonlinear.



TABLE II-4

RUDDER COMMAND AND RESPONSE

<u>Figure</u>	<u>Curve</u>	<u>Rudder</u> <u>Command(deg)</u>	<u>Initial</u> <u>Condition(deg)</u>
II-5	1	+30.0	-30.0
II-5	2	+25.0	-25.0
II-5	3	+20.0	-20.0
II-5	4	+15.0	-15.0
II-5	5	+10.0	-10.0
II-5	6	+ 5.0	- 5.0
II-6	1	+30.0	0.0
II-6	2	+25.0	0.0
II-6	3	+20.0	0.0
II-6	4	+15.0	0.0
II-6	5	+10.0	0.0
II-6	6	+ 5.0	0.0



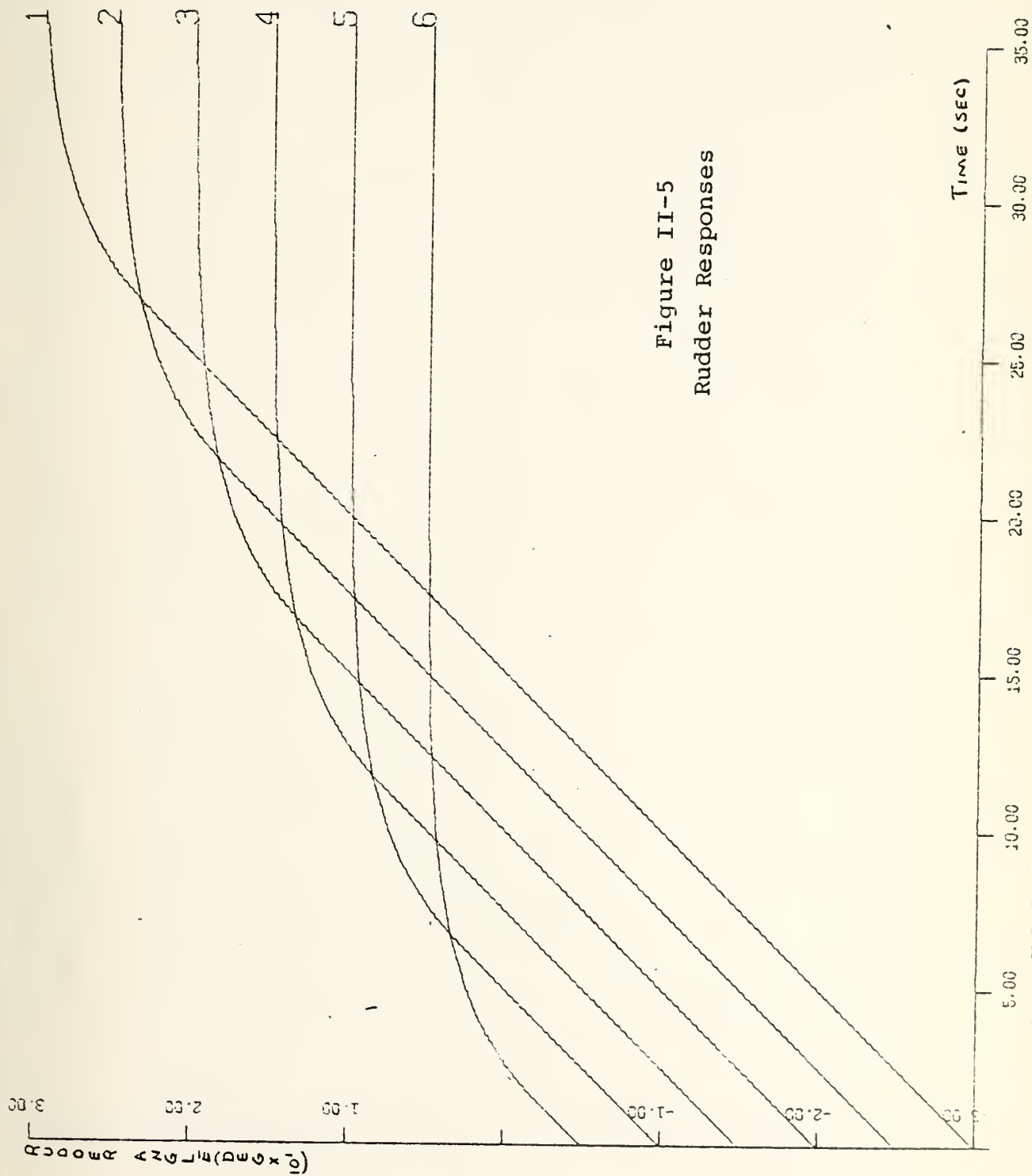
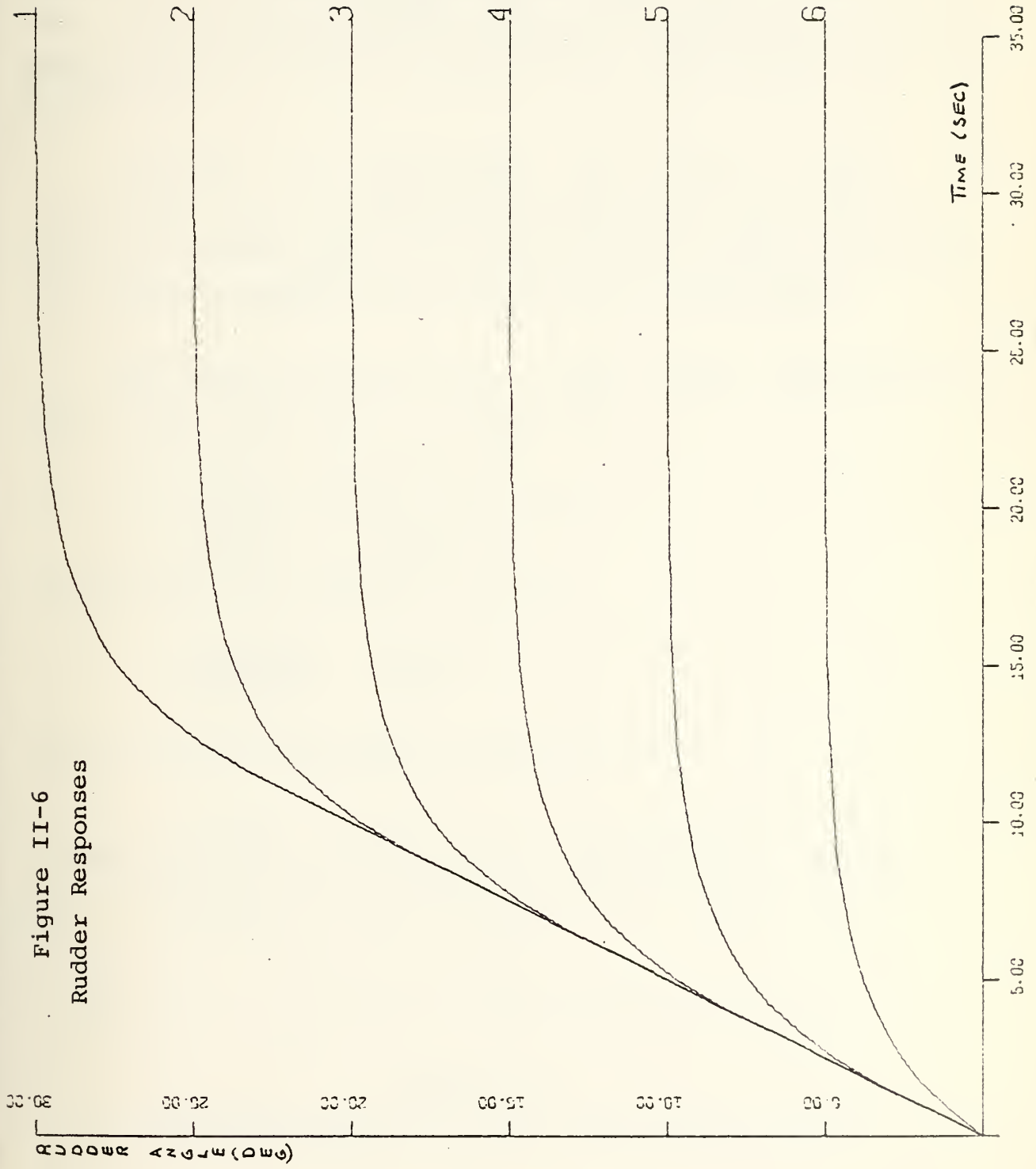


Figure II-5
Rudder Responses

Figure II-6
Rudder Responses



C. ENGINE RESPONSE

Figure II-7 portrays a complex model of a gas turbine propulsion plant^{[5][6]}. This model contains the elements required for a complete dynamic study of the system. For the purpose of this thesis, such a complicated model is not required if the overall input-output relationship can be established.

Reference 5 establishes an output speed (U) relationship for a step input of desired speed (U_d) and is redrawn as figure II-8. The relationship appears to be that of a first or second order system with a time delay.

The system equations for a first order approximation with a time delay may be written as:

$$\begin{aligned} \text{SPDIN} &= K * \text{SPDDES} * e^{-Ts} \\ \text{SPDERR} &= (\text{SPDIN} - \text{SPDCUT}) * G \\ \text{SPDCUT} &= \int \text{SPDERR} \, dt \end{aligned}$$

Which yields the transfer function:

$$\frac{\text{SPDCUT}(s)}{\text{SPDIN}(s)} = \frac{G * K * e^{-Ts}}{s + G}$$

Which is block diagrammed in figure II-9.

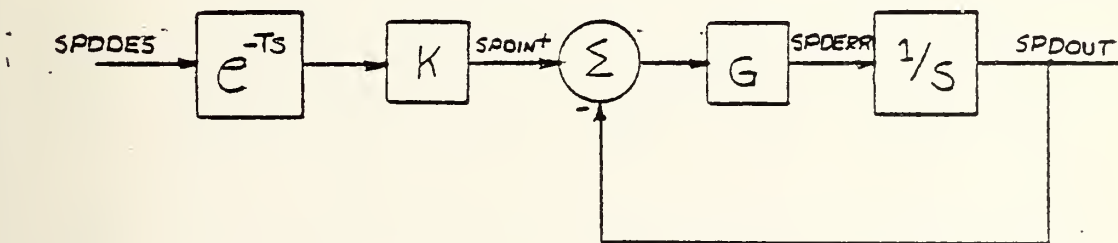
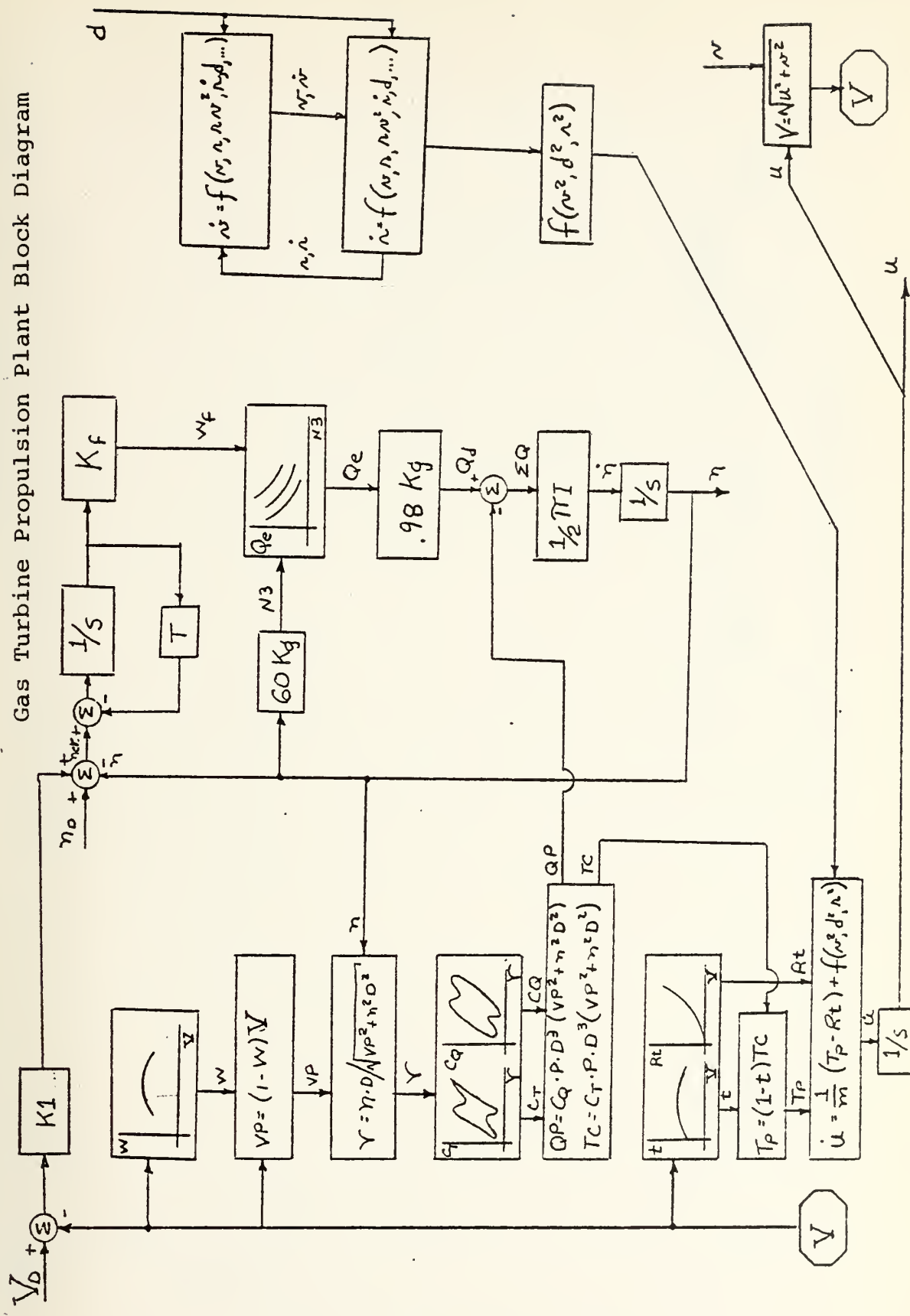


Figure II-9

Propulsion Plant Low Order Model Block Diagram

Figure II-7

Gas Turbine Propulsion Plant Block Diagram



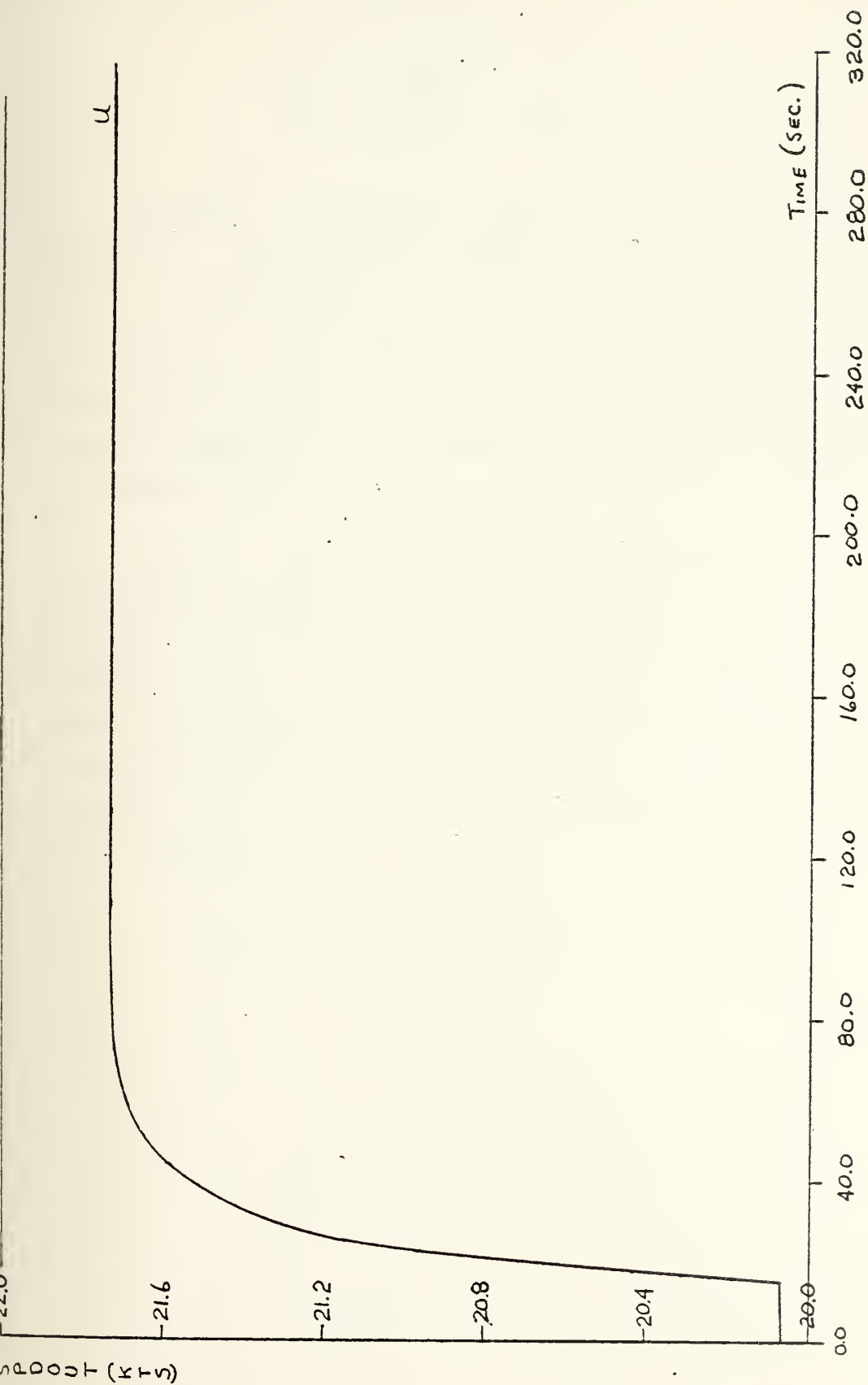


Figure II-8
Propulsion Plant High Order Model Step Response

From figure II-8, the time delay, system gain, and time constant can be estimated as:

$$T = 4.88 \text{ sec}$$

$$K = 0.9877$$

$$G = 0.092$$

Computer Program #3 was used to obtain the step input response. The original complex system output and the low order approximation are compared in figure II-10. As indicated in this figure, the two responses are very close. Considering the linearized approximations made in the equations of motion, this response is accurate enough for system study use and is used as the model for speed control in chapter III.

Similar methods may be used to obtain simplified low-order models for other high-order propulsion systems now in use (e.g. pressure fired boiler systems, 1200 lb. systems, etc.). They may not, however, simplify to a first order approximation suitable for system study. A method of computer determination of low-order models of high-order systems is contained in ref. 7 and may be mechanized for this purpose.

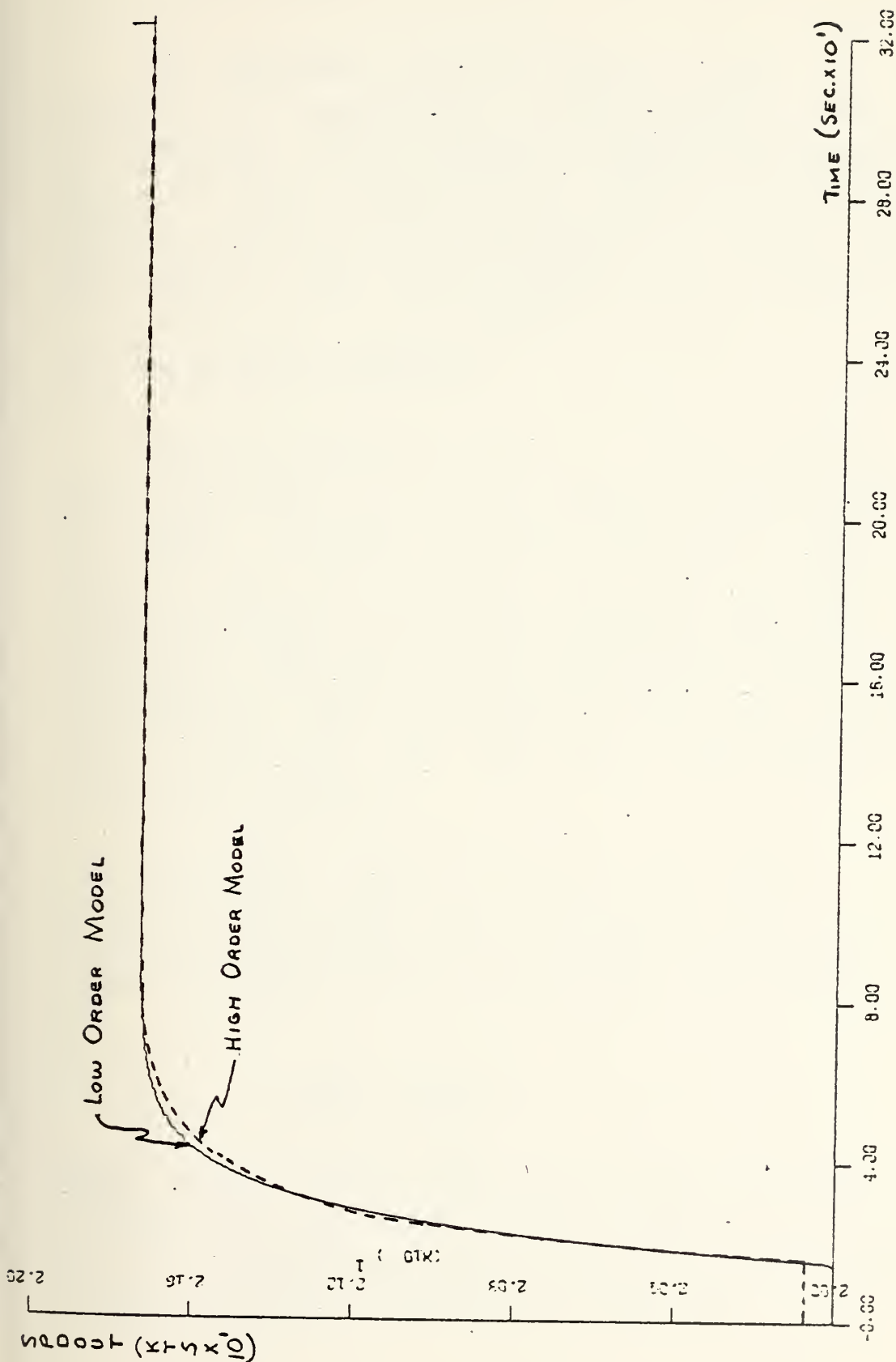


Figure II-10
Step Response Comparison of Low and High Order Propulsion
Plant Models

D. EXTERNAL FORCES

The modeling of ship dynamics cannot be complete without the introduction of external forces which perturb its responses. These forces are caused by many factors and some are more relevant than others in the scope of this thesis. The two that are considered can cause substantial perturbations that must be modeled and eventually accounted for in the control system design.

1. Two Ships in Proximity

Whenever two ships operate in close proximity (less than 250 feet), suction and pressure forces between hulls are present. Studies have been conducted on the Mariner hull[1] which have produced data for construction of a family of curves for two ships passing on the same heading. No data has been found for the cases of two ships not on the same heading. Other restrictions on the work presented in ref. 1 are that the ships are of the same type and of similar hull ratios.

Interactive effects vary as the square of speed. However, this is only true if both ships are at the same speeds. The interaction modification factor is based on the normalized speed of 15 kts. This factor can thus be written as:

$$SPDF = CDOT1^2$$

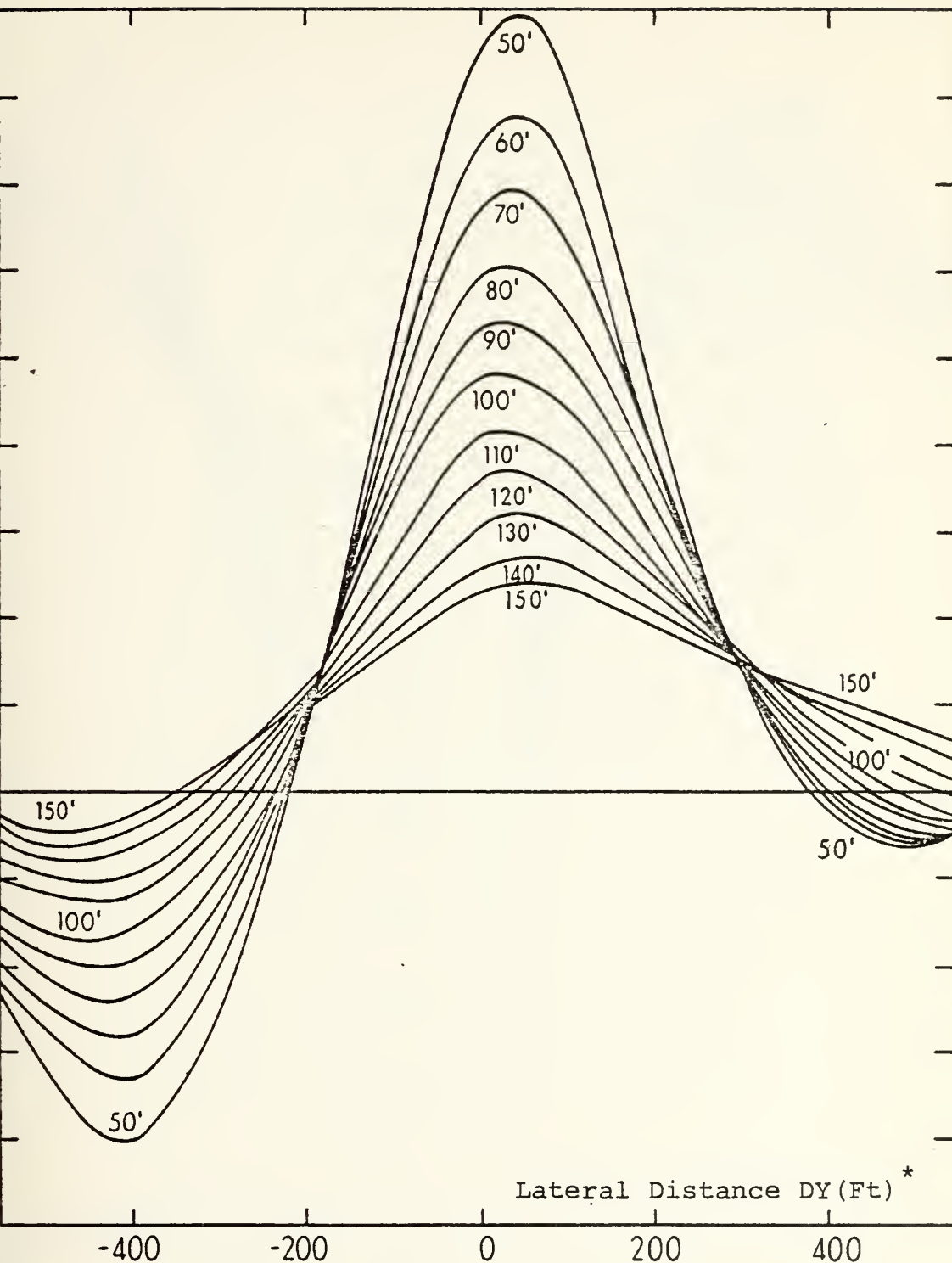
Exact effects on the interaction forces and moments in the situation where the ship's speeds differ are not available. This is inconvenient since during the approach phase, the normalized speed of the approach ship (ship B) can be as

high as 1.5. If the effect on ship B is as stated above but with its own speed causing the interaction modification, the interactive forces and moments can be 2.25 times greater than without speed considerations.

Without the ability to pin down this relationship, it was ignored in the development of the control laws presented in this thesis. Appendix C was written with the expressed intent of illustrating the effect of modifying the interactive forces and moments to the extremum case mentioned above. It must be realized that this case is not considered likely in that it is felt that the interactive forces and moments modification on ship B are more apt to be caused by the speed of ship A. If this is so, since ship A is kept at a constant 15 kts., the interactions need no modification for speed consideration in this thesis.

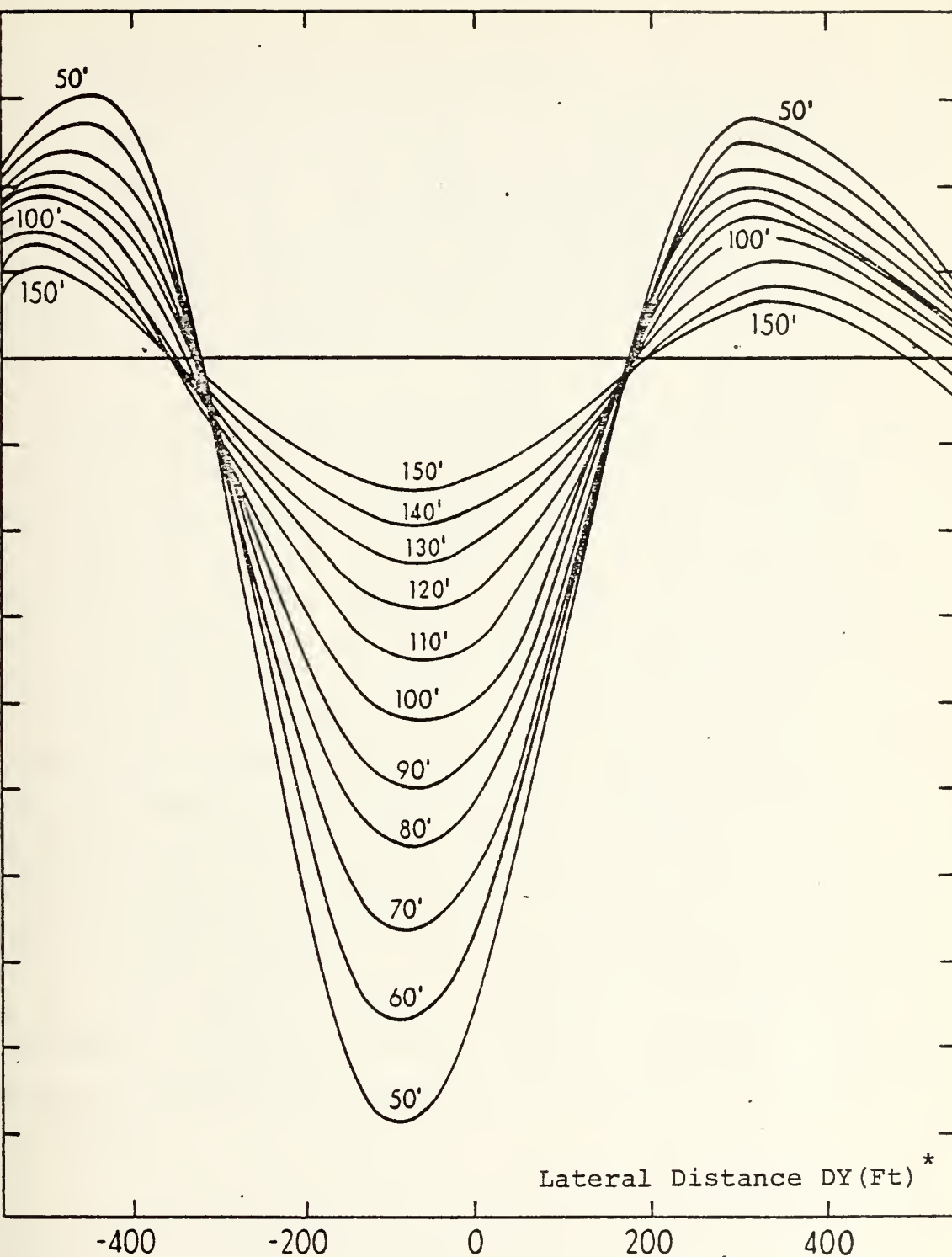
Reference 1 also gives a method of modifying the interactive forces and moments based on different ship lengths. For ease of computation and graphical presentation, the two ships were considered of equal lengths. To modify this to ships of dissimilar lengths, the resulting hydrodynamic derivatives must be modified as shown in ref. 1 (also shown in appendix C).

Since no closed form expression existed for these forces, the family of curves reproduced as figures II-11 and II-12 were quantized in subroutine SLOPES (an adaptation of the subroutine of the same name from ref. 11). (Appendix C contains a curve fitted subroutine that was compiled after the completion of the research on this thesis. It was not used for any design or simulation runs except for those presented in that same appendix.) An interpolation algorithm is used to approximate the intermediate values between quantized values and between the curves of the family.



* Note: To convert to normalized lateral distance DY -
Divide by the ship length L .

Figure II-11
Family of Interactive Y force Curves



* Note: To convert to normalized lateral distance DY -
Divide by the ship length L .

Figure II-12
Family of Interactive N Moment Curves

The main purpose of this subroutine is to compute the interactive forces between ships in the replenishment at sea situation and output the values for perturbation of the control ship only. Even though both ships are affected by these perturbations, a one ship control system which is effective regardless of the other ships motion is considered. Consequently, the interactive forces on the second ship are ignored.

Subroutine SLCPEs is contained in appendix A. Figure II-13 is a geographic plot of the two ships passing at 105.6 feet with their rudders amidships (0 degrees). Figure II-14 and figure II-15 show the magnitude of the lateral force (Y force) and rotational moment (N moment) of the reference ship on the ship making its approach (control ship). The reference ship is at 15 kts. and the approach ship is at 22.5 kts. The control ship starts its approach 5 ship lengths (2639.0 feet) astern and 0.4 ship lengths (211.12 feet) laterally displaced. The most graphic portrait of the effects of these forces and moments appears in the yaw changes which are presented in figure II-16. From these figures it becomes readily apparent that these perturbations cause violent motions of the ship which must be accounted for in any control system development. Throughout the development of such a control system in chapter III, these forces and moments are considered inherent in the model for RAS control.

2. Waves

The modeling of sea state in the form of waves and wave interactions has occupied the time of many researchers [8][9]. The exact formulation of waves will not be accomplished in this thesis. Since the main concern here will be to test the control scheme developed in chapter III, a much simplified wave generator can be used. To introduce the required experimental perturbations on the designed control system a periodic wave system with a fundamental frequency and its second harmonic is used. Some small random wave properties are introduced that ride on these two sinusoids. A simple expression of this combined wave can be written as:

$$W = WF \cdot \sin(WE) + (PI \cdot WF^2 / WL) \cdot \sin(2 \cdot WE) + WF \cdot WRV \cdot \sin(WE)$$

where:

W denotes the Wave

WF is the Wave Force

WE is the Wave Encounter radian Frequency

WL is the Wave Length

WRV is the Wave Random Variable

PI is 3.1415926

With this wave as a basis, a method of modeling this in the dynamic environment of the total ship simulation was defined. The modeling includes the introduction of this wave into the three degree of freedom equations of motion. To accomplish this a set of defining relationships were developed. First the general wave direction is input to establish the direction of the wave encounter on the ship. If the ship direction is YAWDP2 and the wave direction is WD, the expected wave direction is defined as:

$$EWD = WD - YAWDP2$$

Next the wave encounter frequency (radian frequency) can be established with knowledge of the ships normalized true speed (CDCT2), wave length (WL) and normalized wave velocity (WV). The wave encounter frequency (WEF) is then:

$$WEF = 2 \cdot \pi \cdot (CDCT2 + WV \cdot \cos(EWD)) / WL$$

The total wave encounter (WE) is nothing more than the wave encounter frequency (WEF) times time. This gives the wave encounter radian frequency required in the simple expression for the combined wave previously shown.

This does not complete the task, since the individual wave forces of each degree of freedom must be derived for this general wave expression, namely the components of WF. Again a much simplified version of the more complex real life wave forces were used. The X and Y forces are considered first. These can be modeled as cosine and sine functions of the expected wave direction (EWD) such that:

$$WFX = WF \cdot \cos(EWD)$$

$$WFY = WF \cdot \sin(EWD)$$

where WF is the total wave force of the encountered wave.

The rotational N forces are a little more difficult to establish. By considering that no rotational forces are created by a wave directly on the bow or stern, or directly off the beam, and that it is maximum when the wave is at 45 degrees off the bow or stern, a much simplified approximation is developed. Realizing that this method is very crude, the N force can be written as:

$$WFN = WF \cdot \sin(2 \cdot EWD)$$

To add more creditability to the wave model, a random



variable is added to the wave force at the waves fundamental frequency. A gaussian (normal) distribution was chosen with a zero mean and a standard deviation of one-tenth the maximum allowable force of WF. A zero mean signifies that the expected amplitude of the random wave is 0.0, while the standard deviation signifies that 68% of the random waves will have amplitudes less than one-tenth of the maximum allowable force of WF. Also, 94% will have amplitudes less than one-half the maximum allowable force of WF. This small added perturbation allows for verification of the model simulation with a stochastic force, which in turn adds creditability to the developed control systems.

What remains is to define the total wave force (WF). It is important not to fall into a common simulation pitfall which inevitably causes unneeded design changes. A sea state does not increase at an infinite rate. It therefore is incorrect to start a simulation with initial conditions set for calm sea and immediately introduce a high sea state perturbation. The initial large perturbation transient can give results that are not only unrealistic, but can cause the model and control system to produce unstable results. This is especially true in this case since the linearized (small perturbations about an operating point) equations of motion are used.

With this in mind, a ramp feed in of the wave force with a limiter at the desired maximum wave force (WFMA) was used. The slope of the ramp was established to impart minimum initial transients, yet increase the wave force to an acceptable testing level within the time frames of the simulations of chapter III. The slope is designed such that the maximum wave force is reached in 94.815 seconds actual simulation time (4.548 seconds problem time).

Computer program #4 was used to verify the wave action



model. Table II-5 on page 56 indicates the figures produced and changes in input wave length (WL) and wave direction (WD) for each run. The input parameters that were constant for all runs are tabulated below:

$$YAWDF2 = 0.0$$

$$CDCT2 = 1.5$$

$$WS = 5.0^*$$

$$WFMA = 0.1137$$

* NOTE: WS is the unnormalized wave speed. Conversion to normalized wave velocity is:

$$WV = WS/15.0$$

Introduction of the wave forces is accomplished by multiplying the established wave forces by the rudder hydrodynamic coefficients for the individual reference directions. This effectively scales the wave forces to the ship model being used. The wave force result is coded in the ship simulation program as follows:

$$IF12 = RA1 \cdot D2 + YY2 + KA1 \cdot WY$$

$$IF22 = RE1 \cdot D2 + YN2 + KB1 \cdot WN$$

$$IF32 = RC1 \cdot D2 + NC2 + KC1 \cdot WX$$

Detailed results of the wave force effects are given in chapter III and will not be dealt with here.

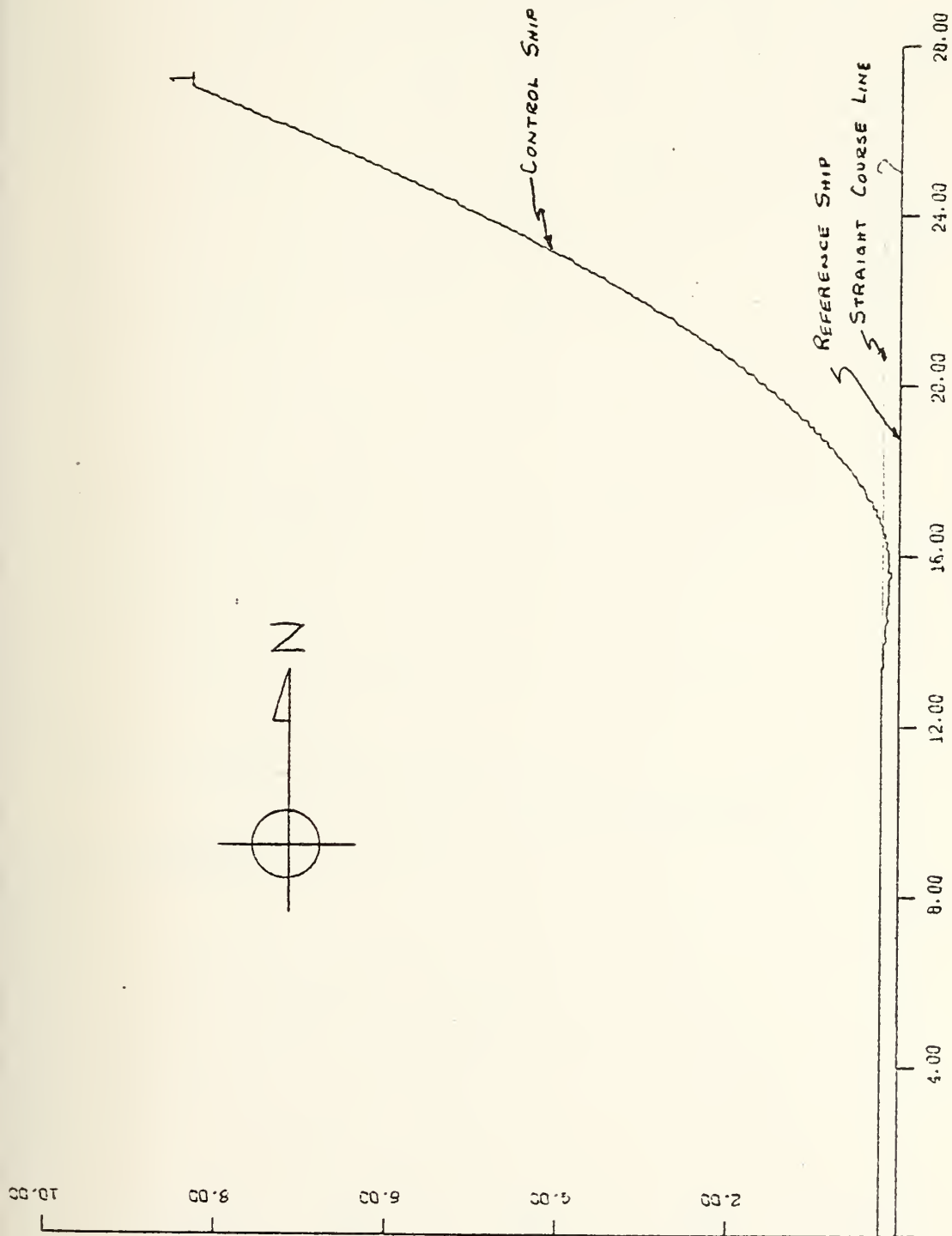


Figure II-13
Interactive Forces Effect on the Geographic Plot

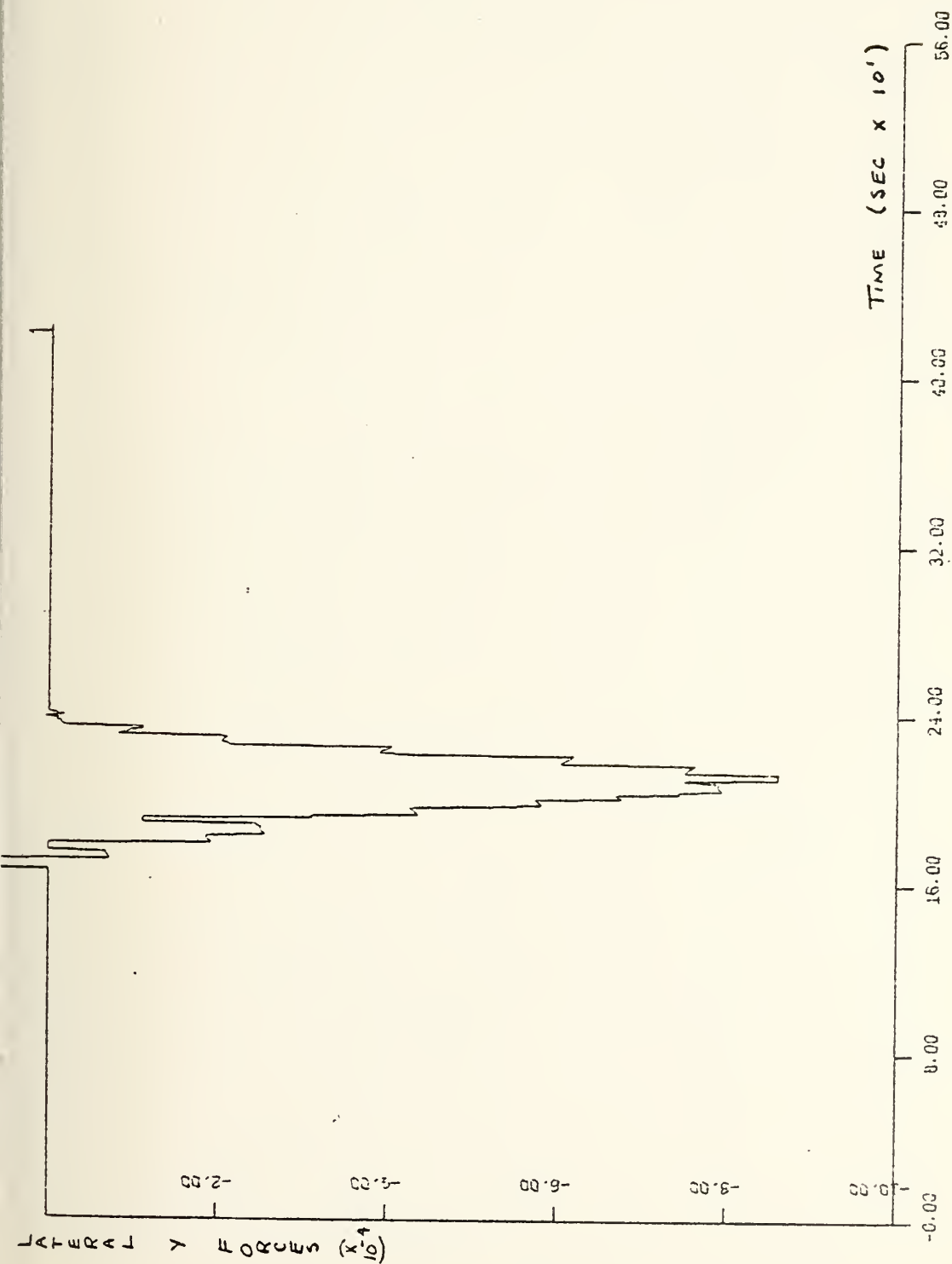


Figure II-14
Lateral (Y) Forces

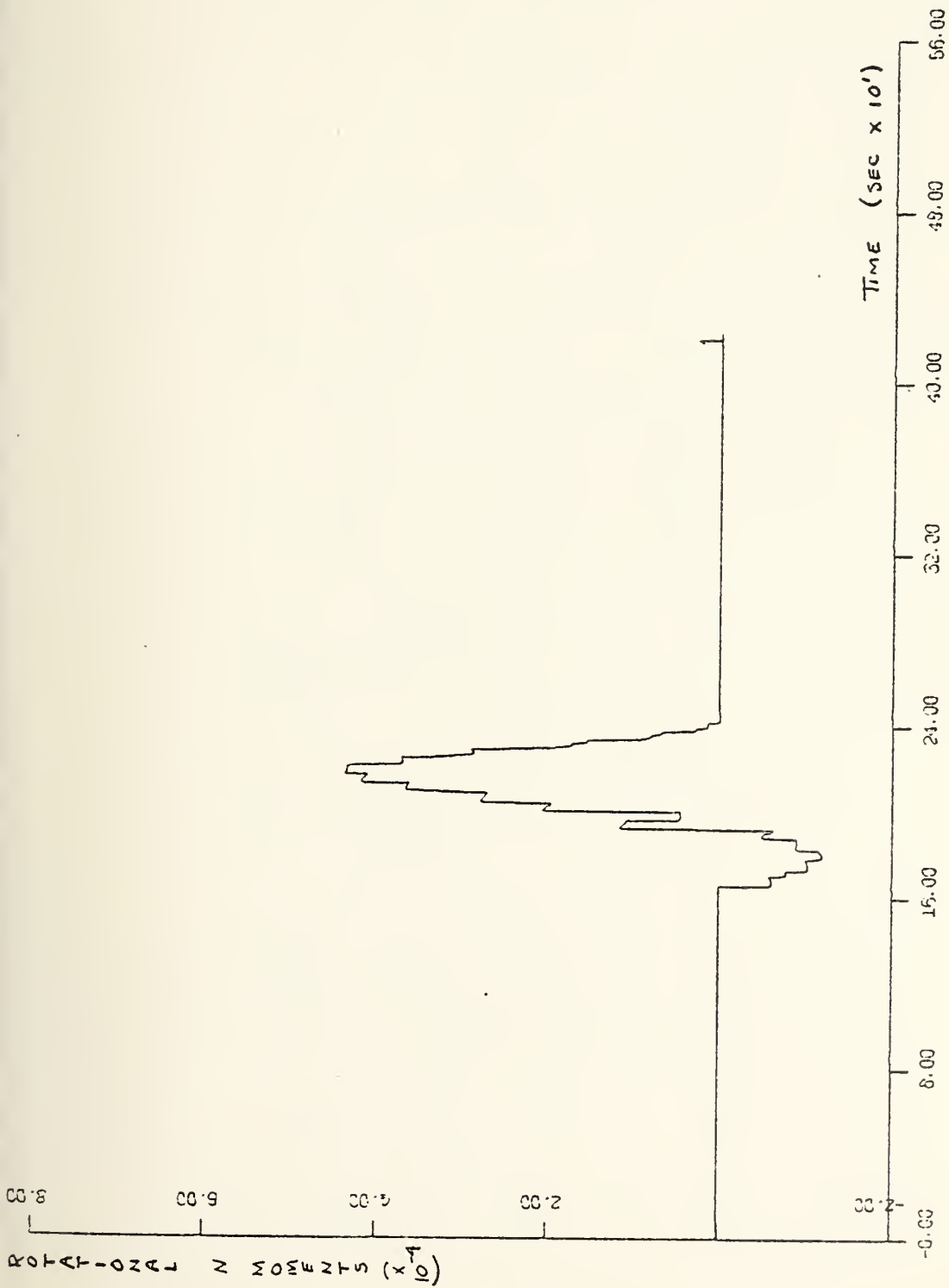


Figure II-15
Rotational (N) Moments

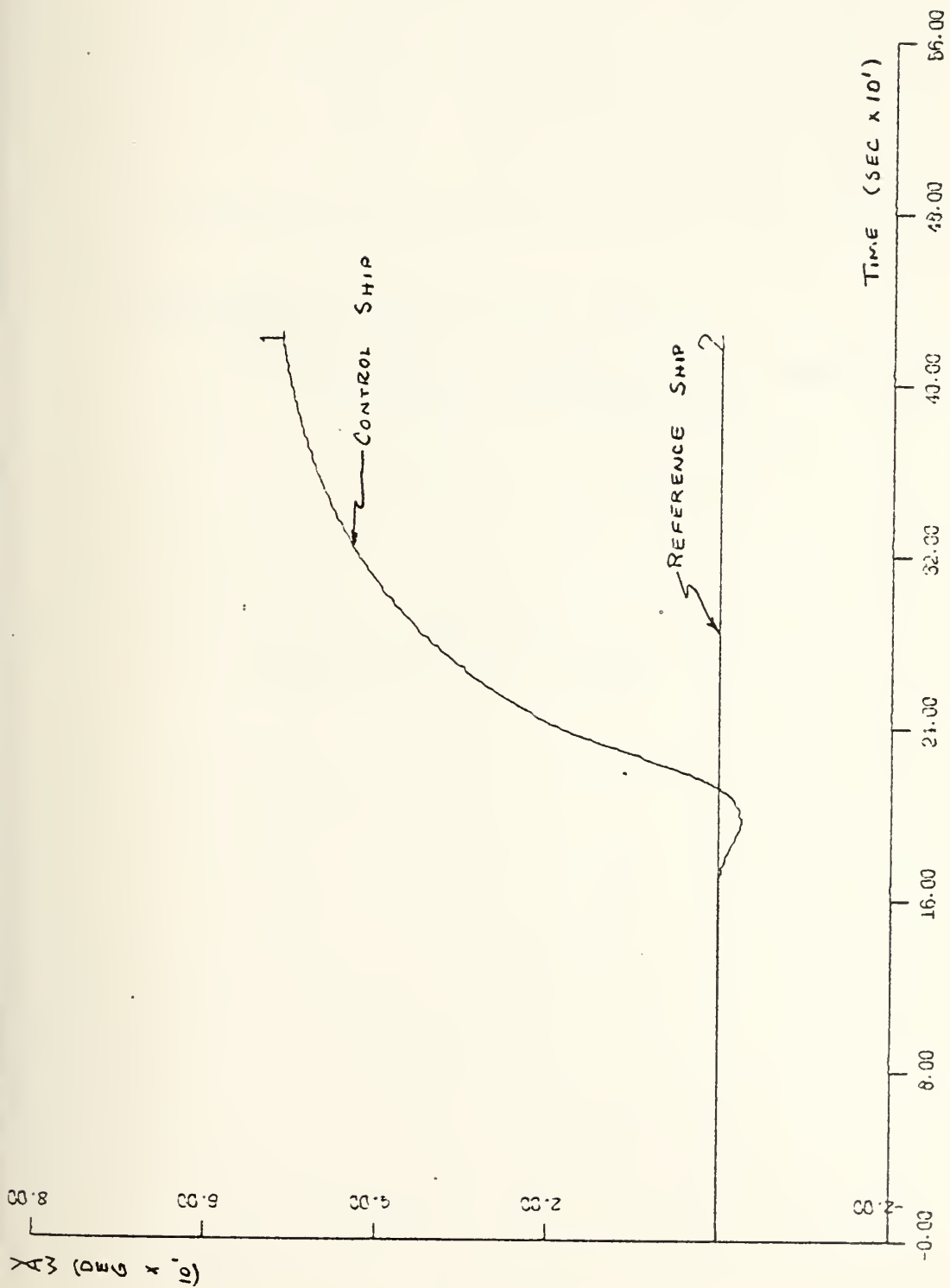


Figure II-16
Interactive Forces Effect on Yaw of the Control Ship

Run	Figure*	Input Parameters	
		WL*	WD*
1	II-17	0.5	015
2	II-18	1.0	015
3	II-19	1.5	015
4	II-20	0.5	030
5	II-21	1.0	030
6	II-22	1.5	030

* NOTE: WL is given in ship lengths
WD is given in degrees
Curve numbers of all runs corresponding to
wave force components are:

Force	Curve
WX	1
WY	2
WN	3

Table II-5
Wave Simulation Listing

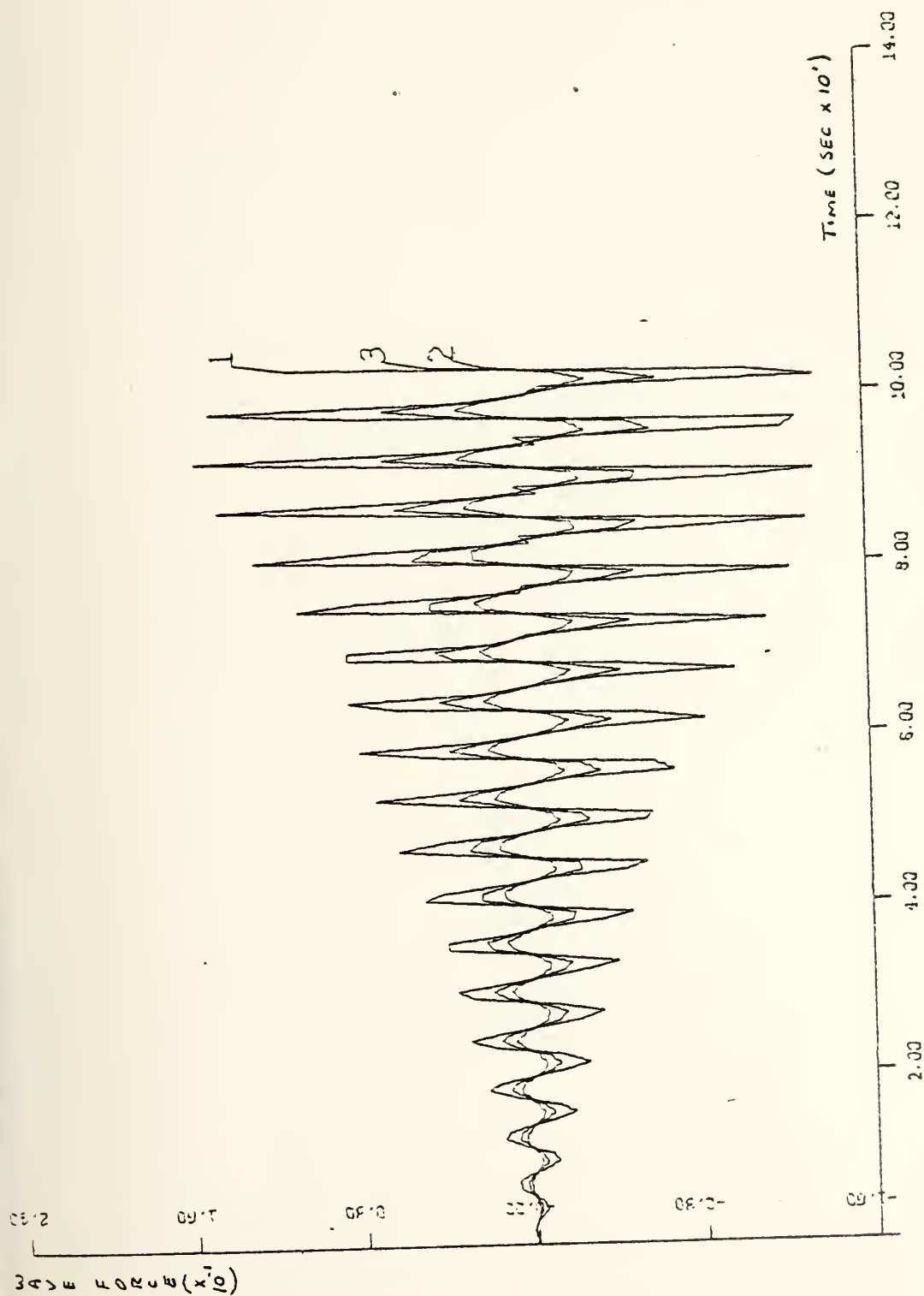


Figure II-17
Wave Simulation Run #1

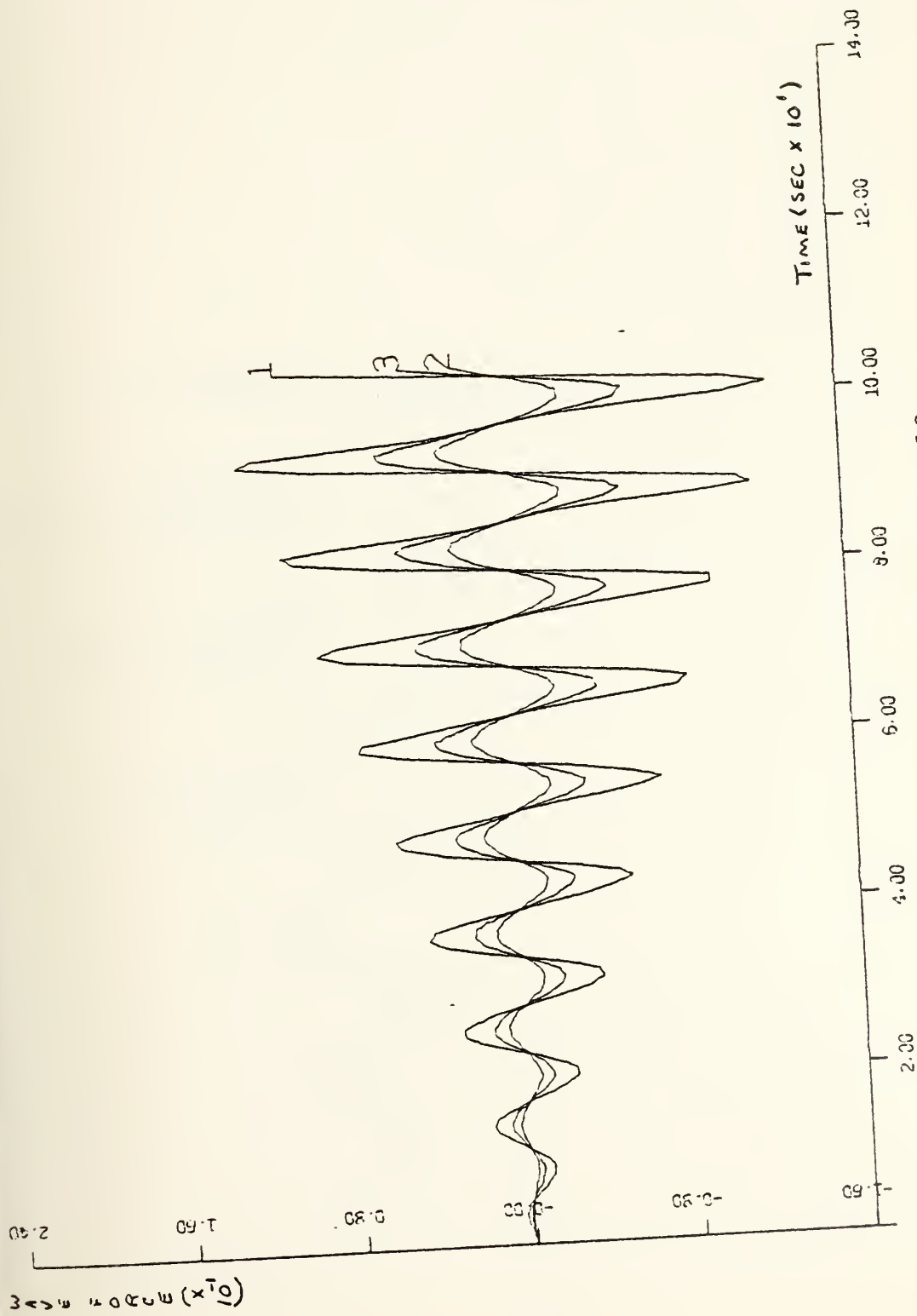


Figure II-18
Wave Simulation Run #2



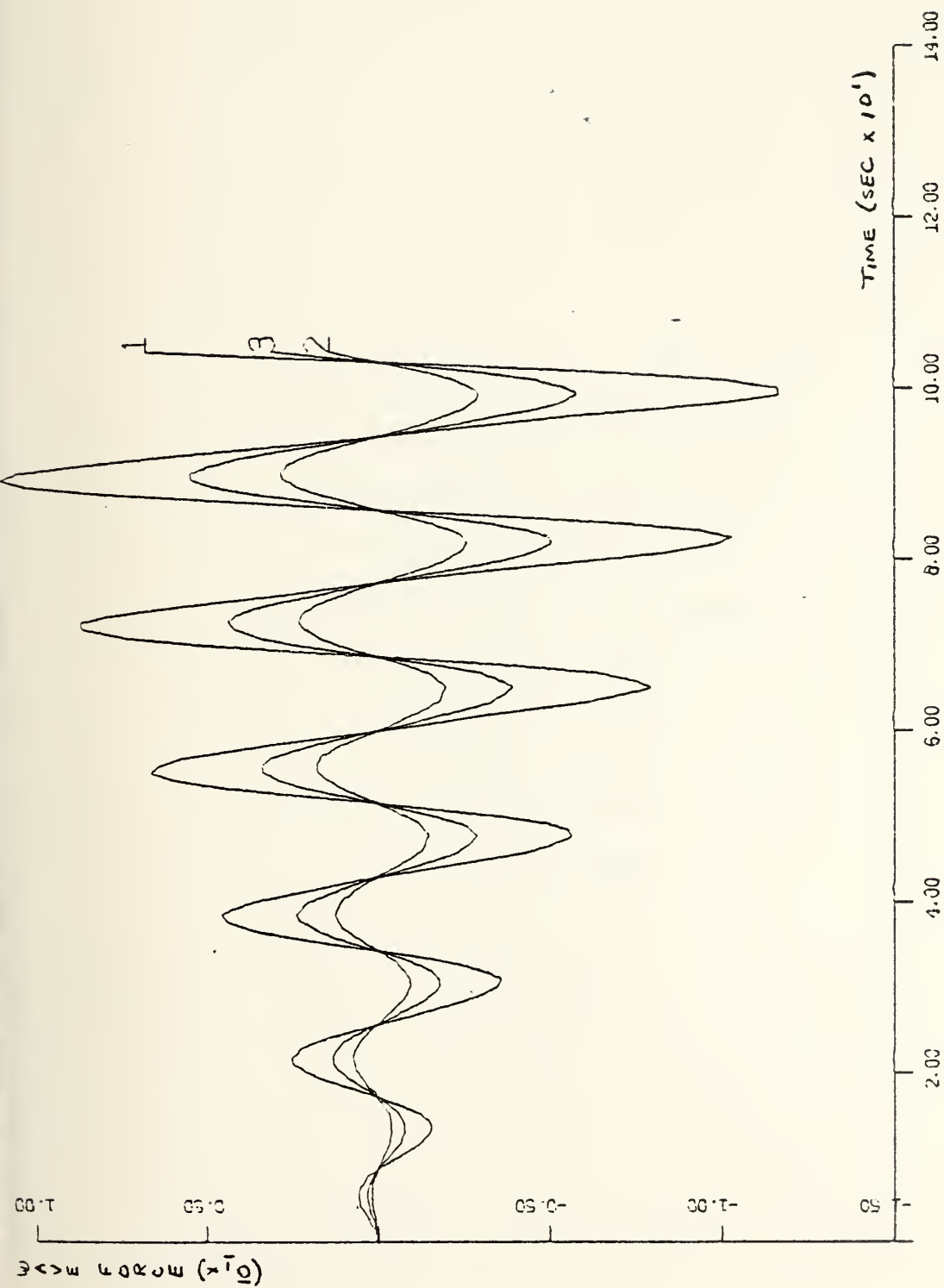


Figure II-19
Wave Simulation Run #3



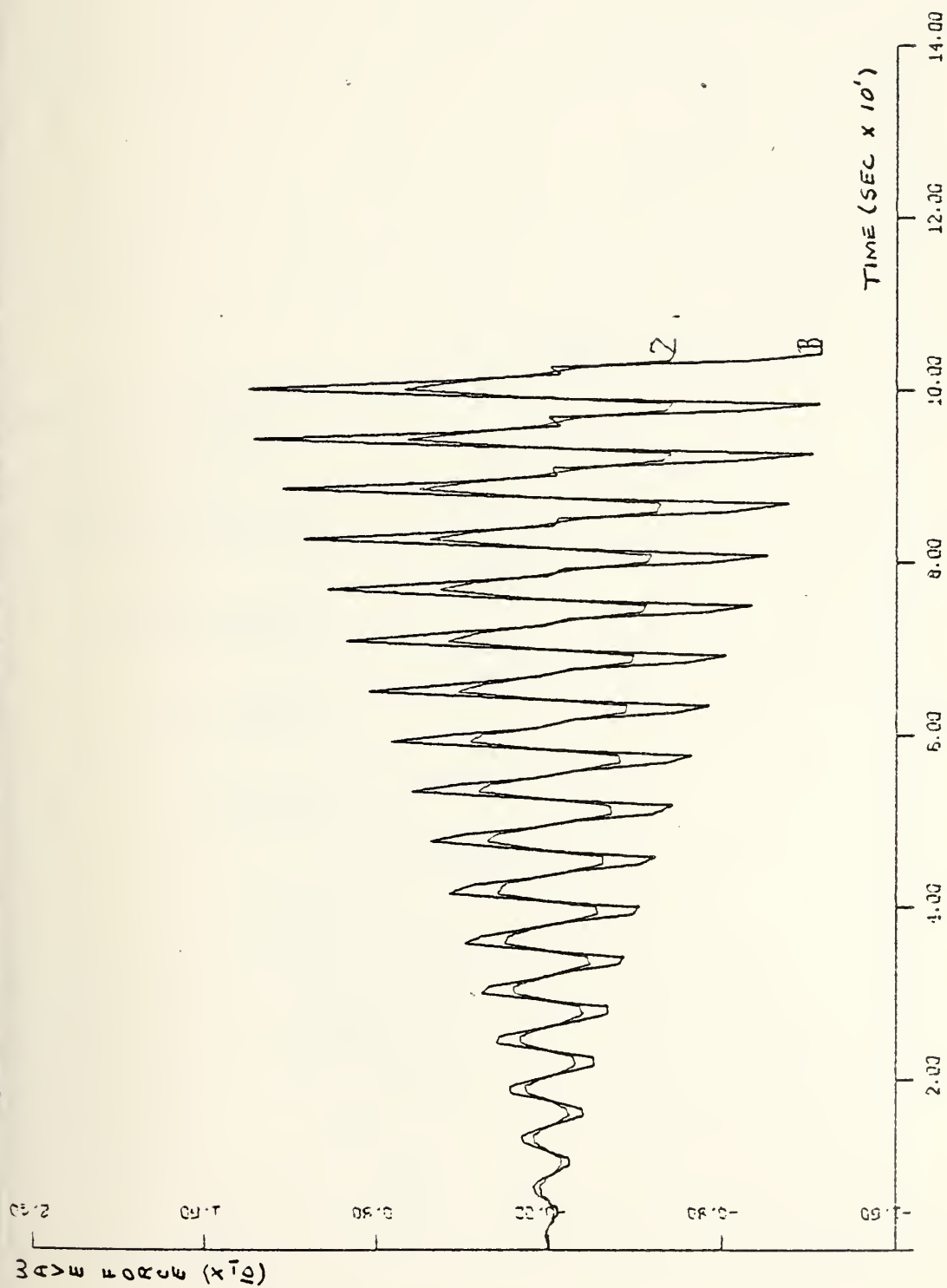


Figure II-20
Wave Simulation Run #4



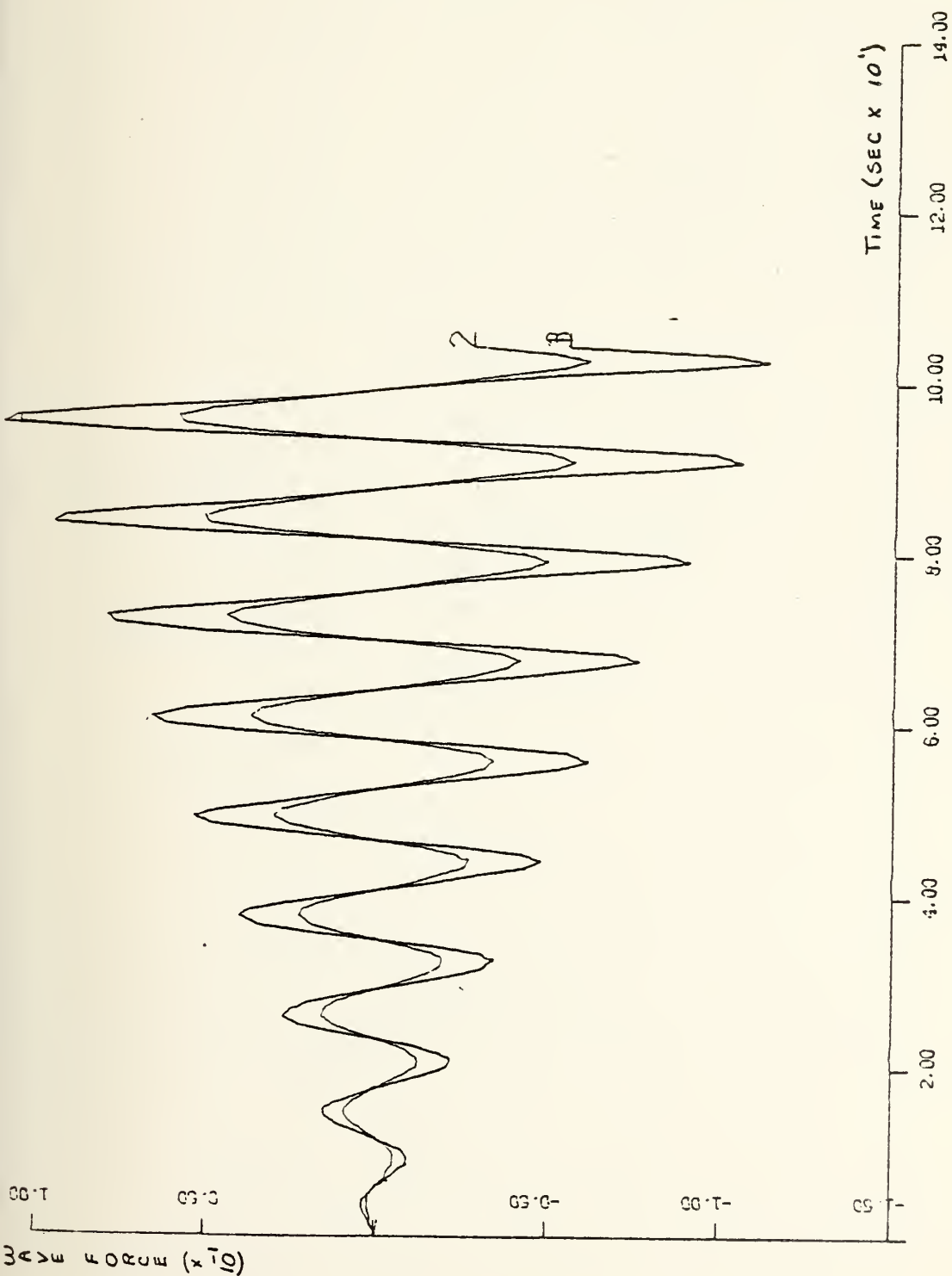


Figure II-21
Wave Simulation Run #5





Figure II-22
Wave Simulation Run #6



III. REPLENISHMENT AT SEA

A. HEADING CONTROL

1. Control Choice

Many studies involving replenishment at sea (RAS) have treated the problem as a multivariable system^{[10][11][12][13]}. Academically, there is nothing wrong with this approach. However, as a practical system it leaves much to be desired. The key drawback in the multivariable system is the inescapable dependency on a command and control link between the replenishment ships. The unreliability of UHF communications at these close distances is a much experienced phenomenon. It is felt that any knowledgeable commanding officer would not entrust the safety of his ship to such a questionable link. An alternate method which is described here is a modern extension of the long trusted "seaman's eye" concept, where the sensors and control devices must be self contained on the ship making the approach (hereafter referred to as the receiving ship or ship B).

In all present day RAS operations, the ship on which the approach is being made (hereafter referred to as the supplying ship or ship A), must maintain the replenishment course and speed. The receiving ship accomplishes the maneuvers to maintain station relative to the supplying ship.

The parameters which are presently measured "visually" are relative position (in both the X and the Y directions),

relative head (usually in reference to ordered replenishment course), and relative motion in the X direction (for speed matching). These parameters are usually visualized by the conning officer who in turn gives corrective orders to the helmsman. The helmsman must then translate these verbal orders into rudder and speed commands through the helm and lee helm consoles. The accuracy of the execution of the conning officer's orders is extremely dependent on the ability of the helmsman and throttlesman. This system can be quite effective, and it can also be quite disastrous. This fast reacting and constantly changing environment lends itself to breakdown in communications and manifests the inability of some individuals to cope with the required critical man-machine interfaces.

To eliminate these problems, present state of the art digital computers and sensors are available for immediate implementation of a completely automatic ship control system. Such a control system may be installed on individual ships and be used for RAS without the requirement of having the matching installation on the other ship of the hockup (another drawback of the multivariable approach).

2. Control Method

One of the many pitfalls that may be encountered in digital simulation is the reality of the parameters that are measurable in the real world situation as opposed to those that are incidentally available in the simulation. With this fact as a keynote, Subroutine RBMEAS (Range and Bearing MEASurement) was developed. This subroutine, as listed in appendix A, defines the forward (FWD) and after (AFT) relative and true bearings, and ranges from the receiving ship to similar points on the supplying ship. Figure III-1 delineates the terms used in the subroutine. The SDFn terms

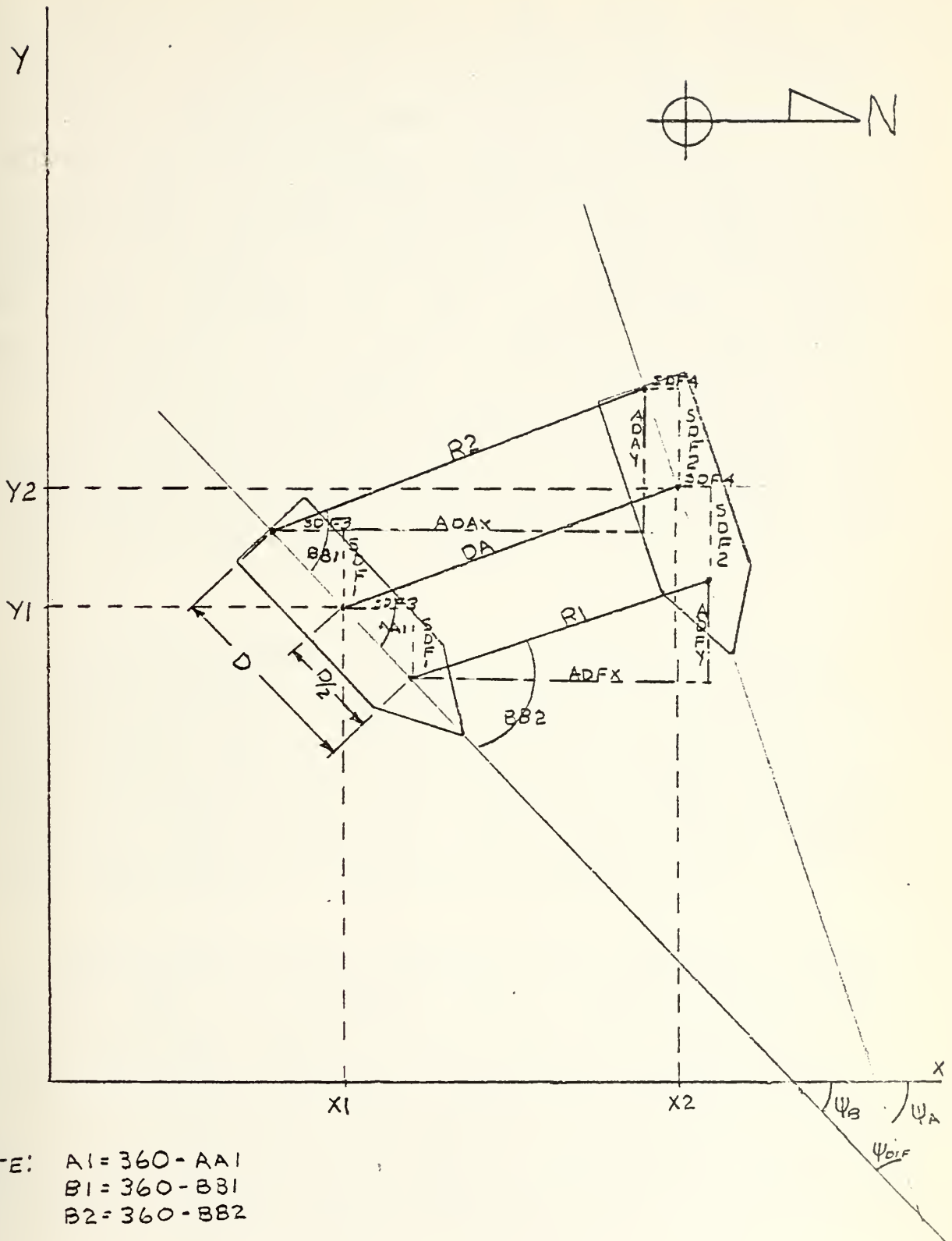


Figure III-1
Measurement Techniques

are used to position the bow and stern sensors and reflectors on ship B and A, respectively, in geographic coordinates as a function of ships head. The FWD distance on the X coordinate is ADFX and the Y coordinate is ADFY. Similarly, the AFT distances are ADAX and ADAY. R1 and R2 are the FWD and AFT ranges measured by a highly accurate ranging device installed on ship B. This same ranging device, if properly provided with a pinpoint reflector on the supply ship(ship A), will give accurate relative bearings FWD and AFT., B1 and B2 respectively. The distance between sensors may be varied, but as a rule should be kept as far apart as possible to allow maximum sensitivity. The distance used in this thesis is 1.0 (one ship length), and the distances were considered the same for both ships. This is not a necessary condition and may be changed to suit the situation.

Subroutine RBMEAS assumes highly accurate sensors in both range and bearing measuring ability. Such sensors are presently available in the form of Radar altimeters[14] and Laser ranging devices. Another possibility for a measuring method is a single sensor time sharing to obtain range and bearing to both reflectors from a single device. Such a single sensor scheme is sketched in figure III-2.

Once the FWD and AFT parameters are available, they may then be used to determine other desired quantities. Subroutine HDGRAS (Heading control for RAS) was developed to output the desired heading corrected for heading difference of ship A and B and the projected correction for distance error. This subroutine is listed in appendix A. The center range and bearing are the average of the FWD and AFT range and bearing output from Subroutine RBMEAS.

The additional heading due to distance is projected as if ship E maintained its present course until it was

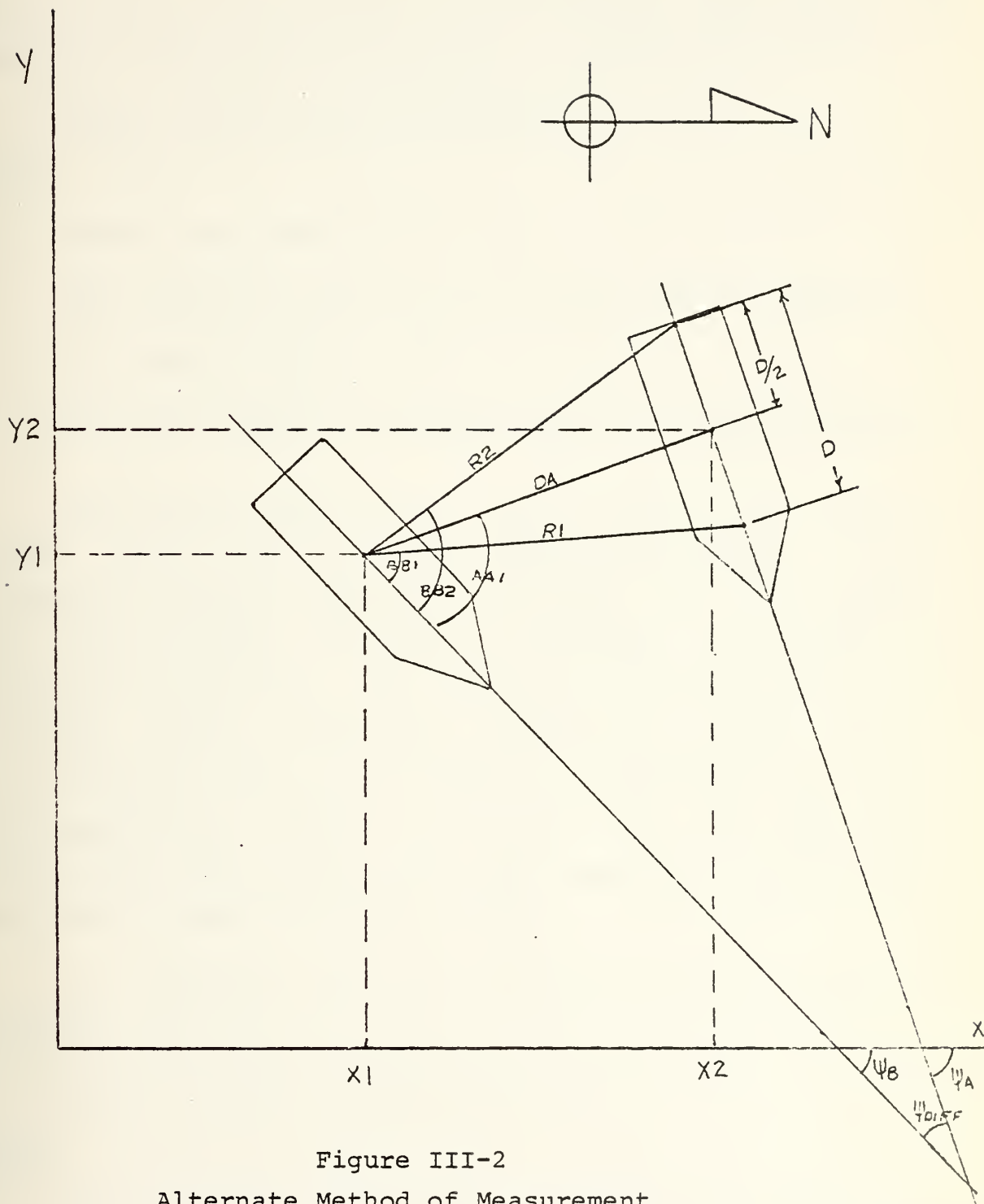


Figure III-2
Alternate Method of Measurement



perpendicular to the center of ship A. The reasoning behind this is illustrated in figure III-3. If the present course will cause ship B to arrive on the station desired (DS), no heading change is required. The expression for PSIADC (Ψ Additional heading due to Distance Correction) is:

$$\text{PSIADC} = \text{RSENS} \cdot (\text{DDC} + \text{DA} \cdot \text{SIN}(\text{AA1}))$$

where:

RSENS = Range SENSitivity gain

DDC = Distance Desired Corrected for side of approach

DA = center Distance Absolute (range)

AA1 = 360 degrees - relative bearing of center position

The heading difference of ship A and B is desired since, even if the range when alongside is correct, a large disparity in heading cannot be tolerated. It is realized that some heading difference (crabbing) is necessary to maintain the distance. This crabbing is due entirely to the pressure forces modeled in chapter II. This heading difference is found by computing the difference in the perpendicular projection between the FWD and AFT measurements and finding the arcsin of this difference divided by the distance between sensors. Figure III-4 indicates a sample of this procedure.

The expression for total desired heading is given as follows:

$$\text{PSIDES} = \text{PSIADC} + \text{WTSENS} \cdot \text{PSIDIF} + \text{PSIE}$$

where:

PSIDIF = Ψ additional heading due to heading DIFFerence

WTSENS = WeighTEd heading difference SENSitivity gain



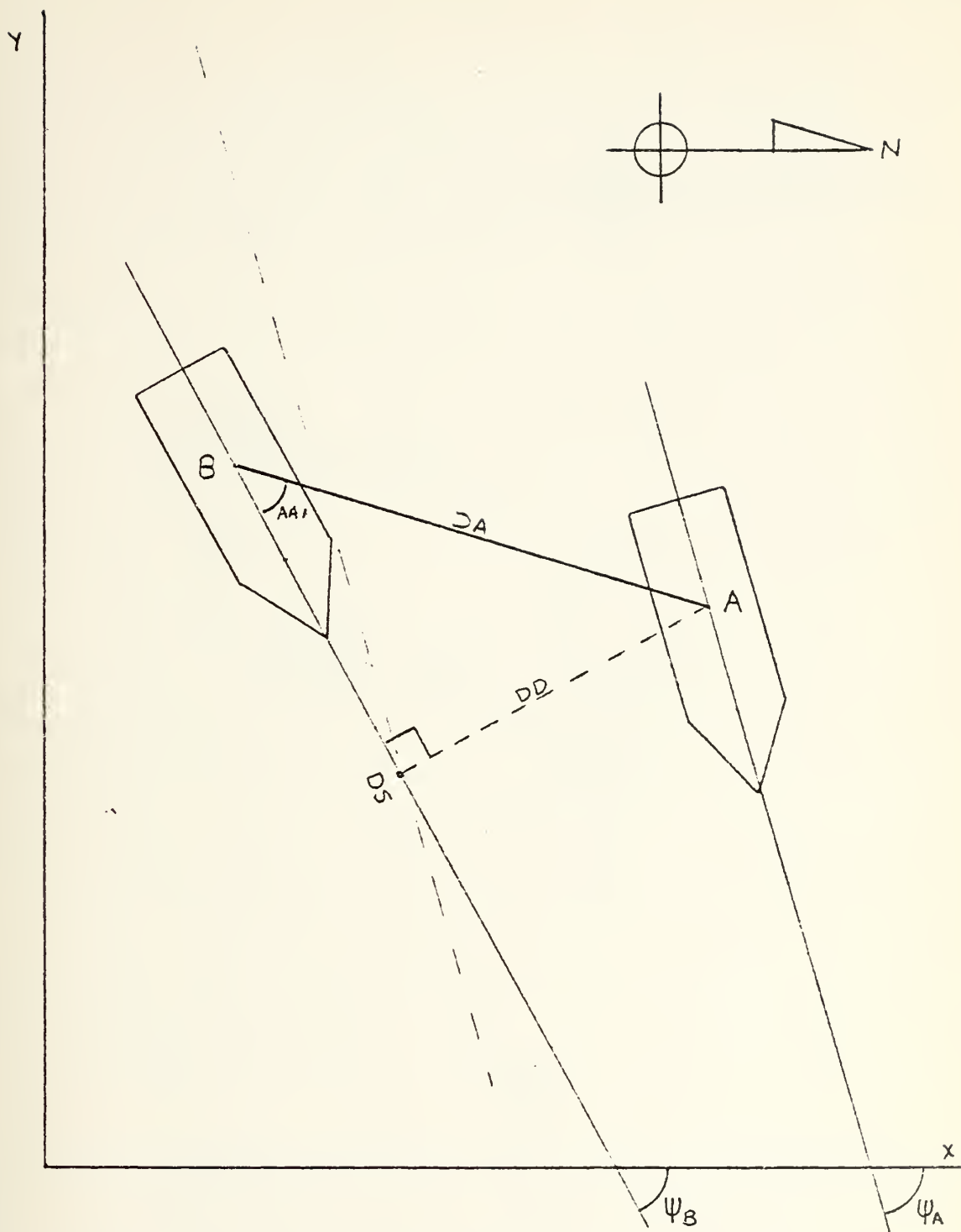


Figure III-3
Distance Logic



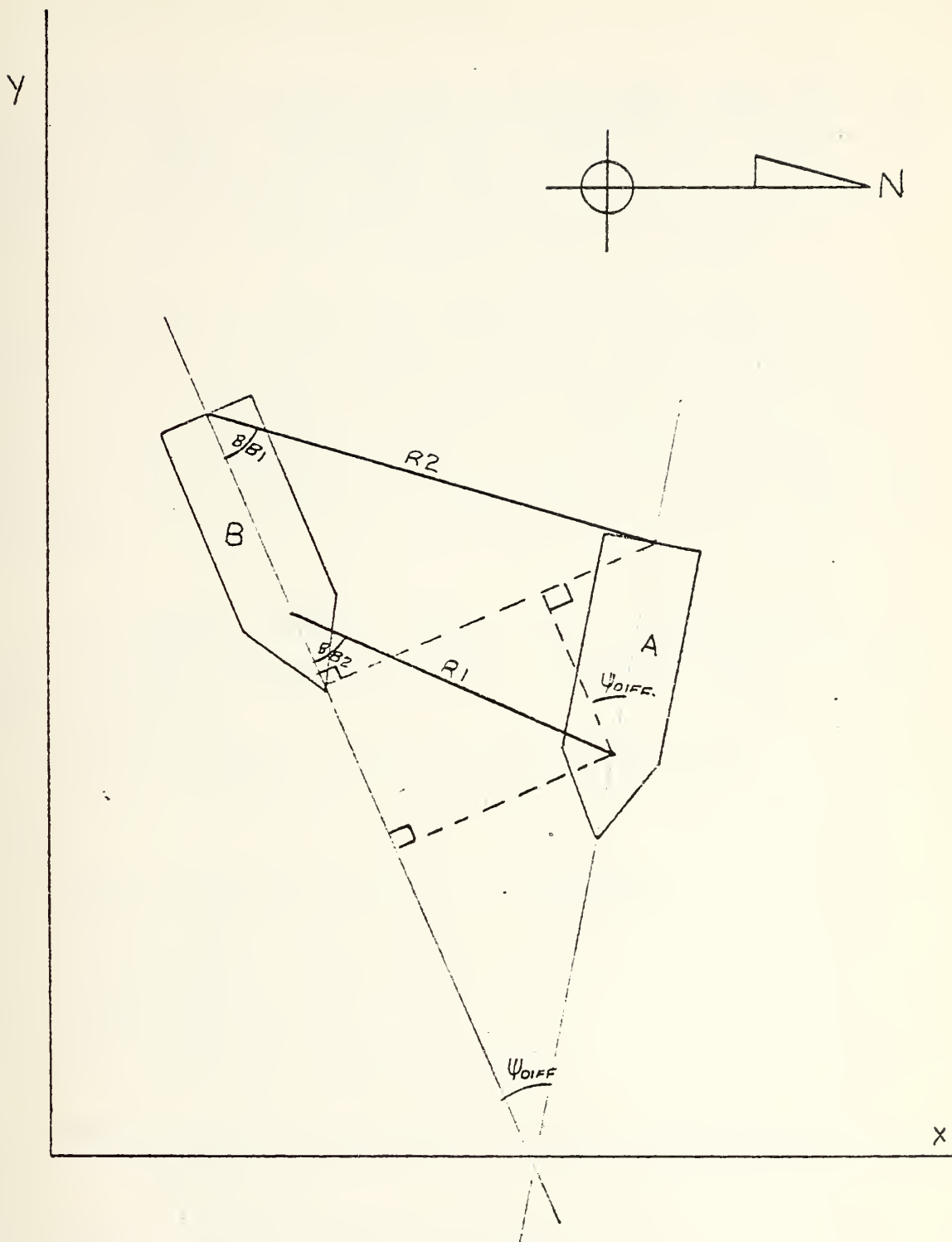


Figure III-4
Heading Difference Calculation



PSIDES = ψ (heading) DESired

Throughtcut the subroutines and main DSL programs, the Function DEGRAD (conversion of degrees to radians and radians to degrees) is used freely. An explanation and listing of this function are presented in appendix A.

The angular velocity of the receiving ship's head is also of concern in the RAS situation. This quantity may be thought of as similar to tachometer feedback in a simple servo control system; and is necessary to damp out the response (the responses of this control system without this feedback is presented in the latter section of this chapter).

The desired rudder command is a combination of the desired heading, angular velocity feedback, and a rudder gain as follows:

$$\text{Desired Rudder} = (\text{YAWD2} - \text{PSIDED} + \text{BDOTFB}) \cdot \text{RGN}$$

where:

YAWD2 = heading of ship B (in degrees)

PSIDED = PSIDES (in degrees)

BDOTFB = VFBG • BDOT2D

VFG = Velocity FeedBack Gain

BDOT2D = angular velocity of ship B heading angle (in degrees/sec)

BDOTFB = angular velocity FeedBack

RGN = Rudder Gain

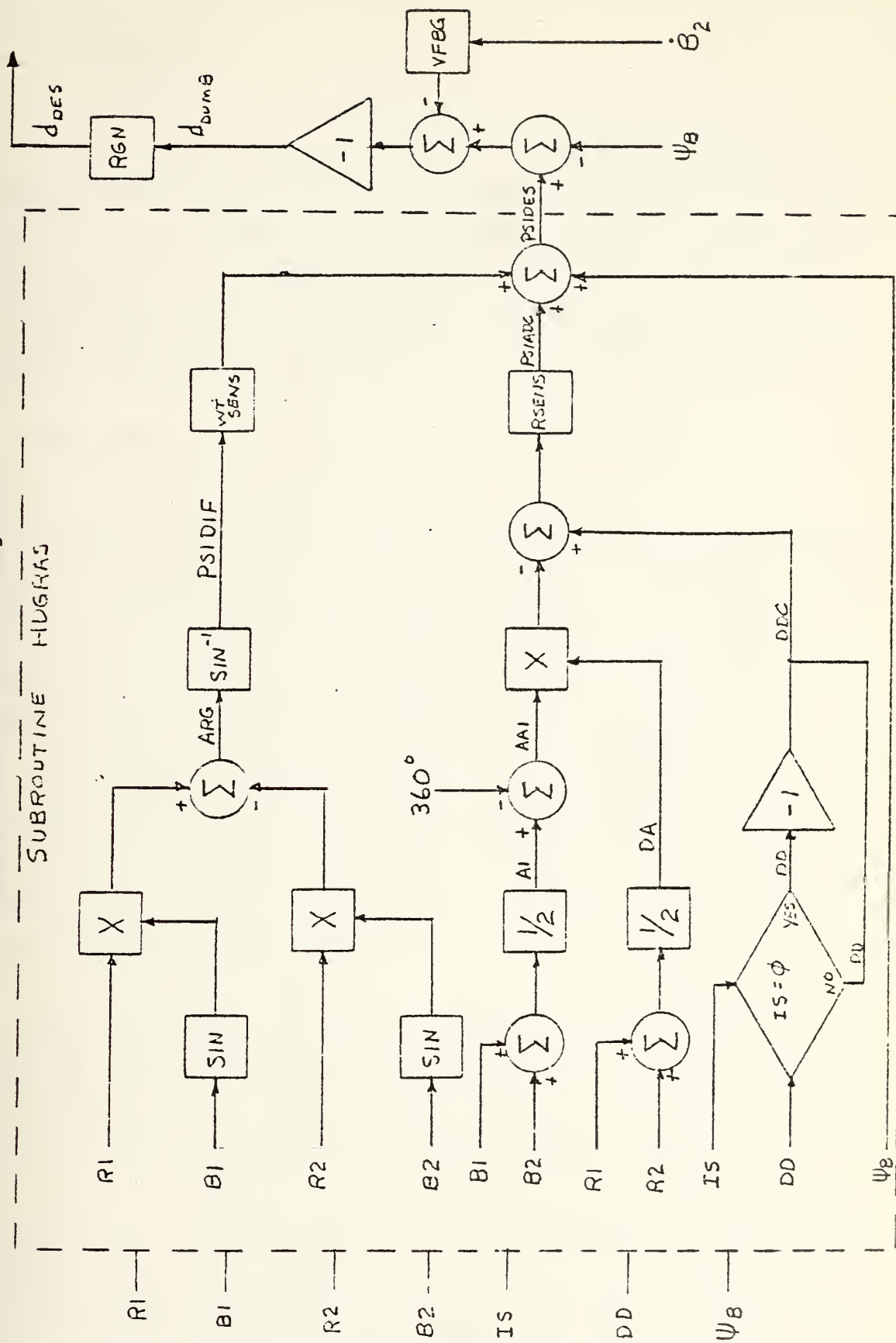
The convention for rudder response dictates negative



rudder as being right rudder, which causes positive yaw. This necessitates making the desired rudder the negative of the forcing function and feedback quantities. The block diagram of figure III-5 presents the control loop from measurement inputs to desired rudder command.

Figure III-5

Rudder Control Block Diagram





3. Optimization

Thus far the control choice has identified four gains (RSENS, WISENS, RGN, VFBG) that must be set for proper position attainment. The nonlinear nature of the system which appears in the form of distance measurements, interactive forces and rudder modeling do not allow for straight forward determination of these gains with normal optimal control theory.

a. Technique

Grossly nonlinear systems require special handling to determine proper gain settings. The method chosen for this purpose was an optimization algorithm developed by M. J. Box (programmed locally as subroutine BOXPLX). This subroutine, listed in appendix A, was used to locate the cost function saddle point in four dimensional space (the dimensions being the previously mentioned gains). The drawback associated with this method is the necessity of iterating the complete nonlinear simulation within function FE for every evaluation of the cost function. The gains sought were found, but unfortunately only after 2 1/2 to 3 hours of CPU time with every 400 iterations allowed.

The mechanics involved in optimizing the chosen cost function include required sub-calculations in many functions and subroutines. Figure III-6 is a flow chart which demonstrates the steps, subroutines and functions required.

Initial optimization was accomplished for one set of initial conditions. By looking at the RAS situation, a probable set of circumstances were envisioned. The scenario setting is the approach phase where the replenishing ship



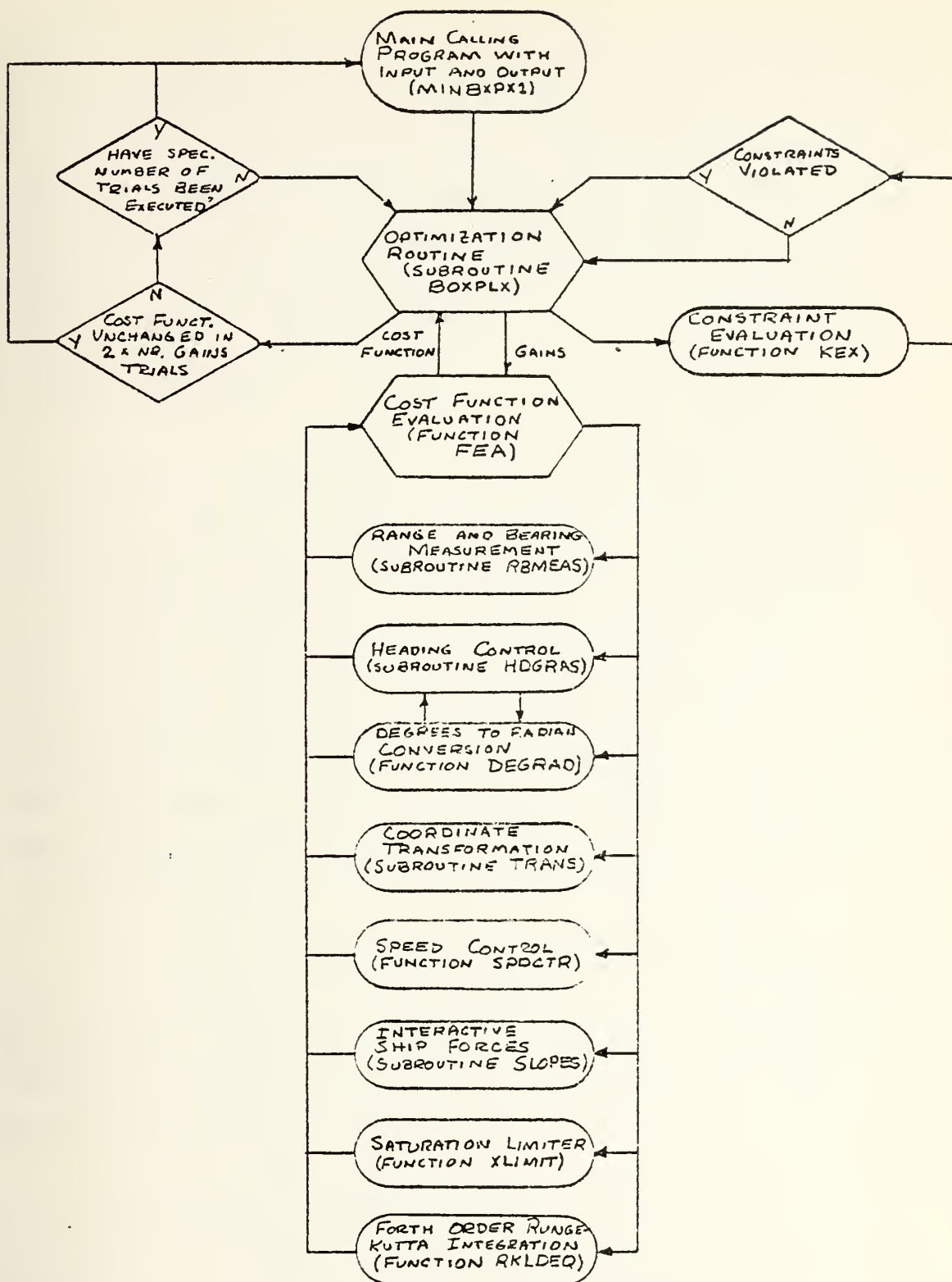


Figure III-6
Optimization Flow Chart



starts a wide approach at 0.4 ship lengths (211 feet) lateral displacement and 5 ship lengths (2639.0 feet) astern of the supply ship. The desired final position is alongside at 0.2 ship lengths (105.6 feet) lateral separation. Both ships have the same initial heading (YAW angle). The supply ship is at 15 kts. (1.0 normalized speed) and the receiving ship makes its approach at 22.5 kts (speed control will be covered later in this chapter).

b. Cost Function

Normal costing of displacement error with the integral of the squared error (ISE) was considered as the optimization tool in subroutine BOXPLX. However, this type of performance measure would weigh the initial displacement error equally with the final position error. This problem can be circumvented by comparing the displacement error to a pre-computed reference track instead of to the desired displacement. For the envisioned scenario, it was conceived that the cost function should weight the distance displacement heavier when the ships are alongside than when the approach is started 5 ship lengths astern.

This was accomplished by using the integral of time times the absolute error (ITAE) as the optimization performance measure. The reference displacement was considered the desired position displacement. The object function can then be written as:

$$CBJ = \int_{t_0}^{t_f} t \cdot |DD - ADY| dt$$

where:

DD = Desired Distance

ADY = Actual Displacement in the Y direction

t = time



A performance measure that is designed to obtain good performance must also take into account other factors besides just position accuracy. Consequently, another cost criterion was decided upon which would also set the gains to reduce the amount of rudder control required when alongside. This particular feature is derived from the desire not to over control with the rudder in such close proximity to another vessel. The inclusion of this term in the performance measure is weighted by unity while the distance accuracy is weighted by a factor of 10.0. This will tend to allow rudder action if the desired position is not maintained. The final approach phase cost function for obtaining optimum gains has the form:

$$CEJ = \int_{t_0}^{t_f} t \cdot (10.0 \cdot |DD-ADY| + 1.0 \cdot |D2|) dt$$

where the additional term is:

$D2$ = rudder response of the replenishing ship

c. Results

In the process of deciding on the best gain definitions previously mentioned, many optimization runs were made. Each set of gains were then simulated in a corresponding DSL program to obtain performance confirmation. Many of these runs did not live up to expectations; causing re-evaluation of the control scheme until the one presented in this thesis was formulated.

Table III-1 shows the input upper and lower limits of search (EU, EL), starting value guess (XS), optimum gain settings (Output) and associated object function value (CBJ) for 20.0 second normalized time simulation run in function FE. These values were then introduced into the DSL simulation program listed as program #5. The results of this simulation are shown in figures III-7 thru III-12. The



Gain	RSENS	WTSENS	RGN	VFBG
BU	2.0	20.0	50.0	10.0
BL	0.1	0.1	1.0	0.01
XS	1.0	1.0	10.0	1.0
OUTPUT	1.86642	2.3869	23.4185	4.35162
OBJ	60.7103			

Table III-1
Approach Phase Optimization Results



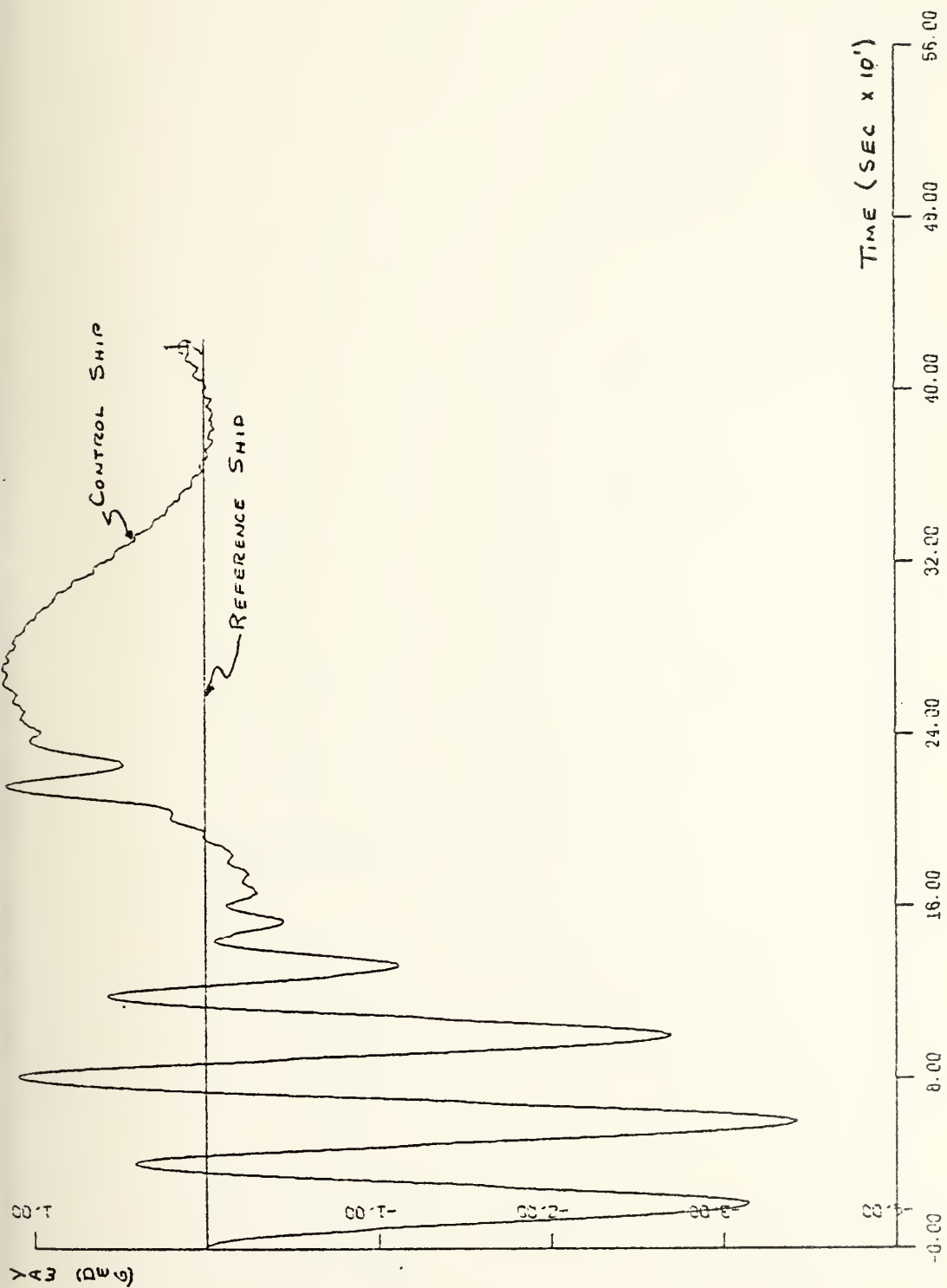


Figure III-7
Approach Phase Yaw Result





Figure III-8
Approach Phase Y Forces

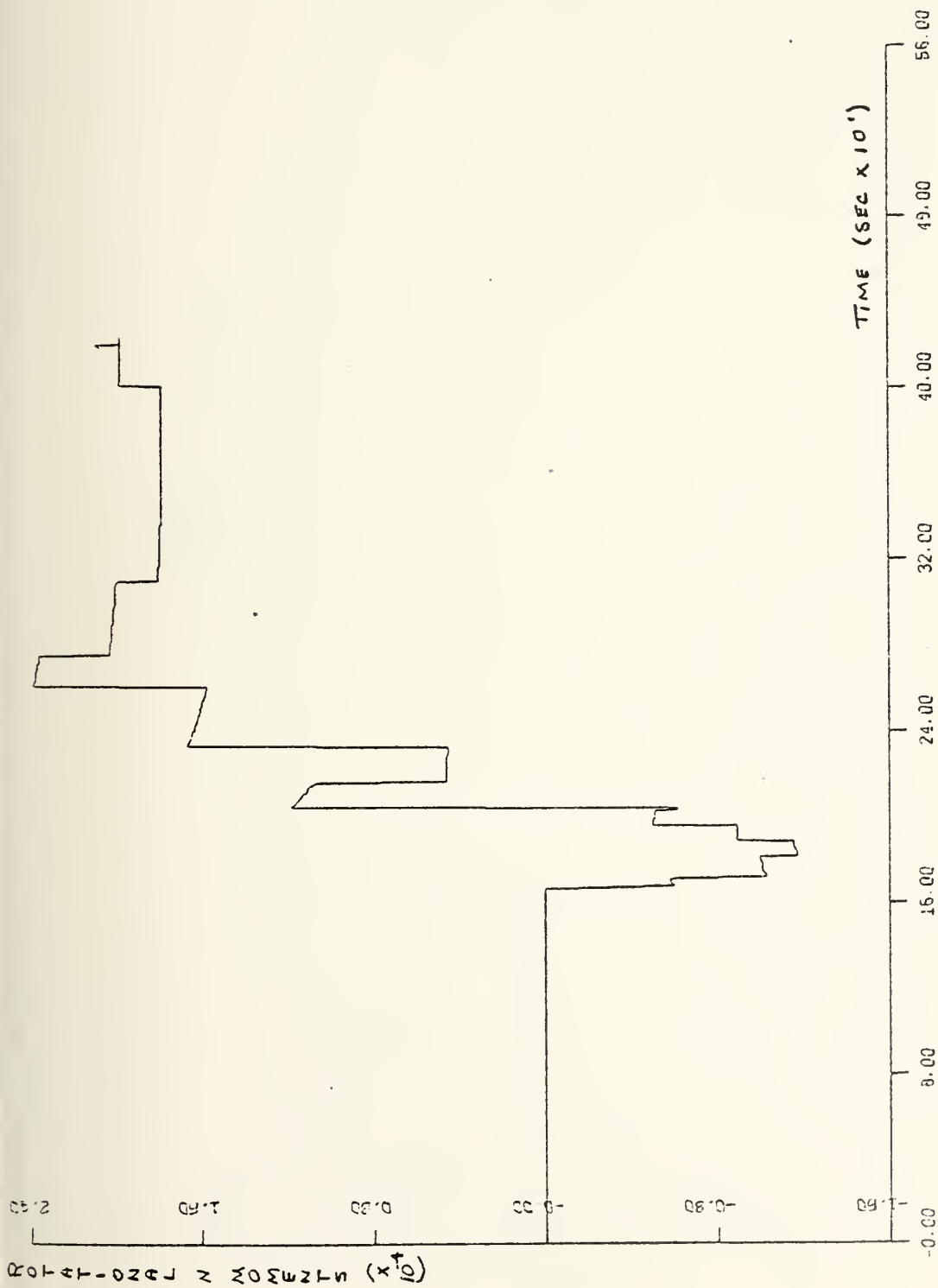


Figure III-9
Approach Phase N Moments

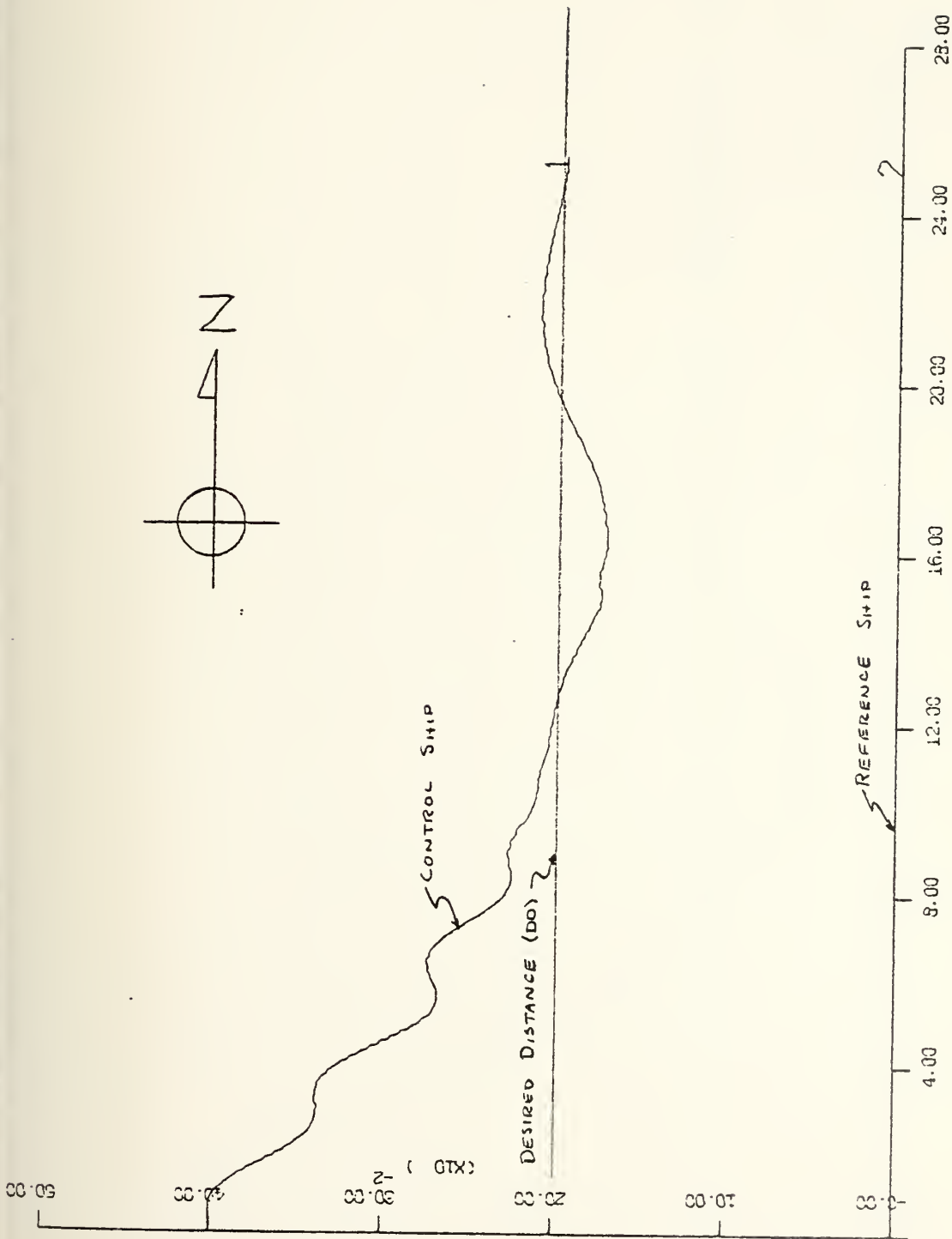


Figure III-10
Approach Phase Geographic Plot

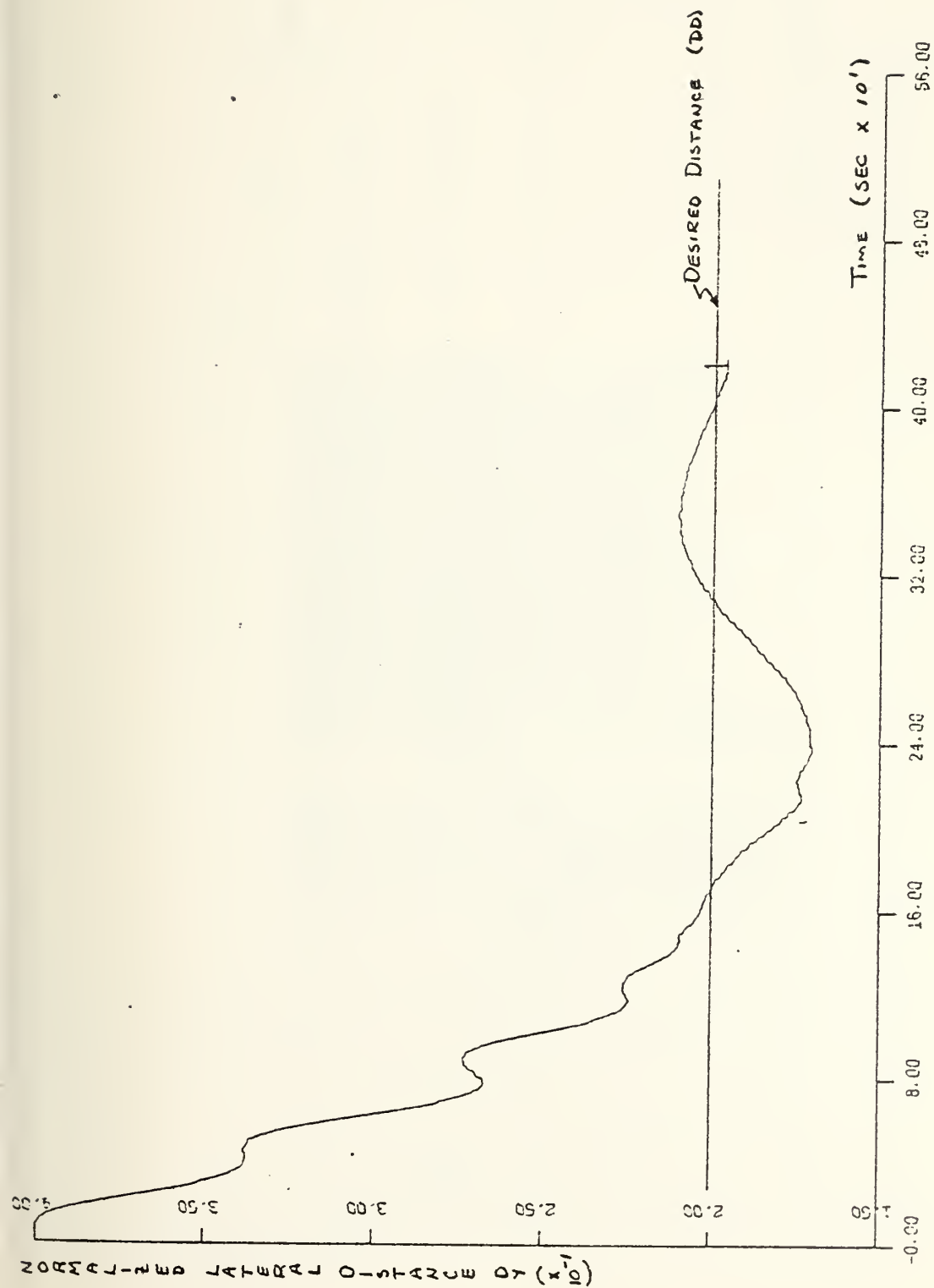


Figure III-11
Approach Phase Lateral Distance DY



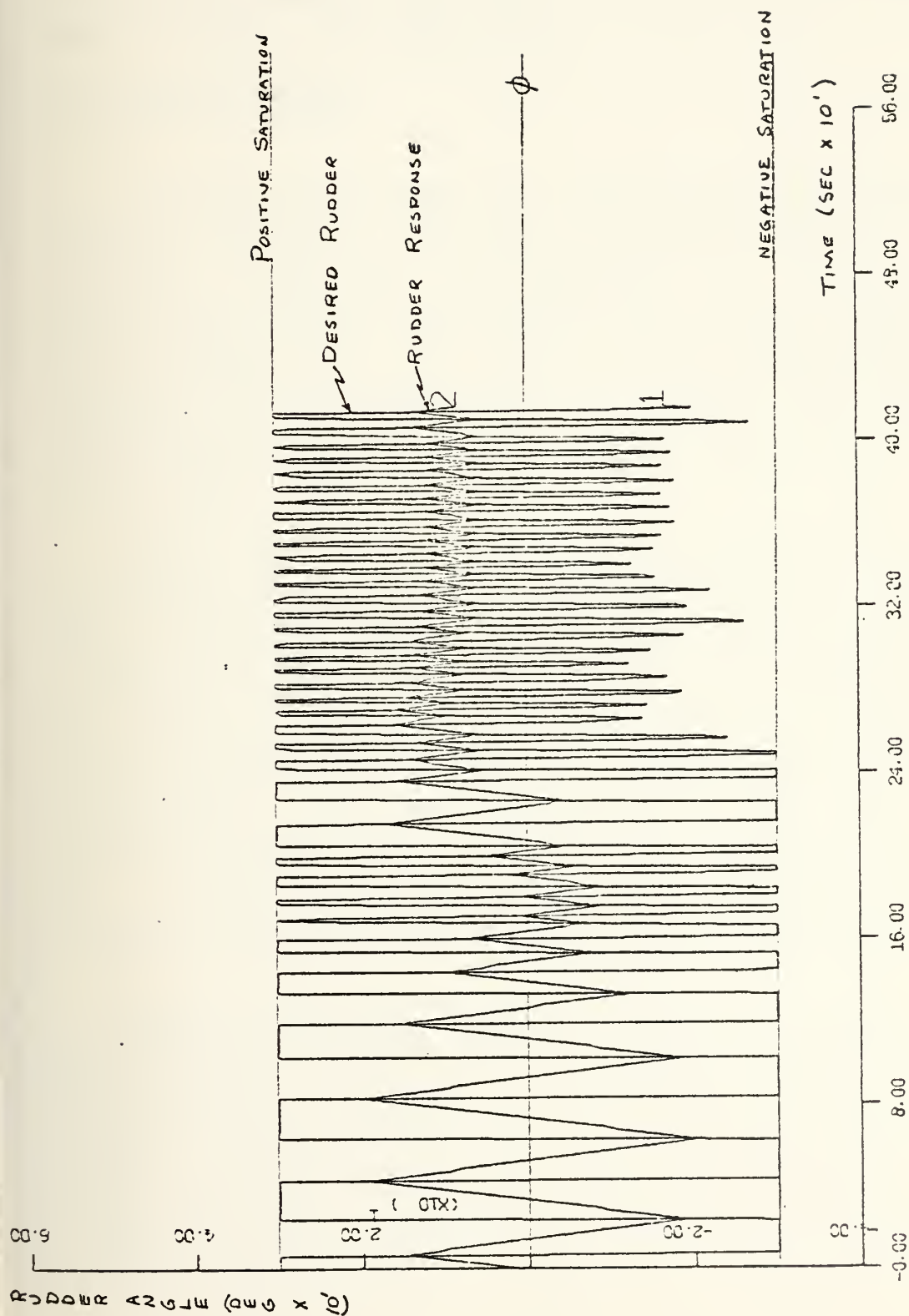


Figure III-12
Approach Phase Rudder Response



geographic plot of figure III-10 indicates excellent positioning in the lateral direction while the rudder response of figure III-12 shows that it settles out to a fairly constant steady state value as the ship settles into its desired position. The time coordinates in all plots are shown in actual full scale time.

d. Control Testing

Now that the "proper" gain settings were obtained, more extensive testing of the control system was required. Three different tests were contemplated: (1) allow a large perturbation turn of the reference ship (supply ship), (2) start approach of the receiving ship (control ship) from different initial conditions of lateral and horizontal displacements, and (3) induce external perturbations in the form of wave forces.

The first test was simulated by turning the reference ship by normal rudder action of figure III-13. This turn with 5 degrees rudder accounted for a total reference yaw change of 15 degrees. The rudder action of the controlled ship shown in figure III-14 was as expected. However, the distance maintainment portrayed in figure III-15 was totally unacceptable. The maximum excursion from the desired distance of 105.56 feet (0.2 normalized distance) was 55.419 feet (0.105 normalized distance). Variances of this magnitude cannot be tolerated in the RAS environment.

Faced with this situation, the tact chosen was to re-evaluate the gains for the new scenario which is called the turn phase. In this phase the initial conditions assume steady state positioning alongside such that the lateral position displacement (DY) is equal to the desired distance [105.56 feet (0.2 normalized)] and that the horizontal position displacement (DX) is 0.0 (alongside).



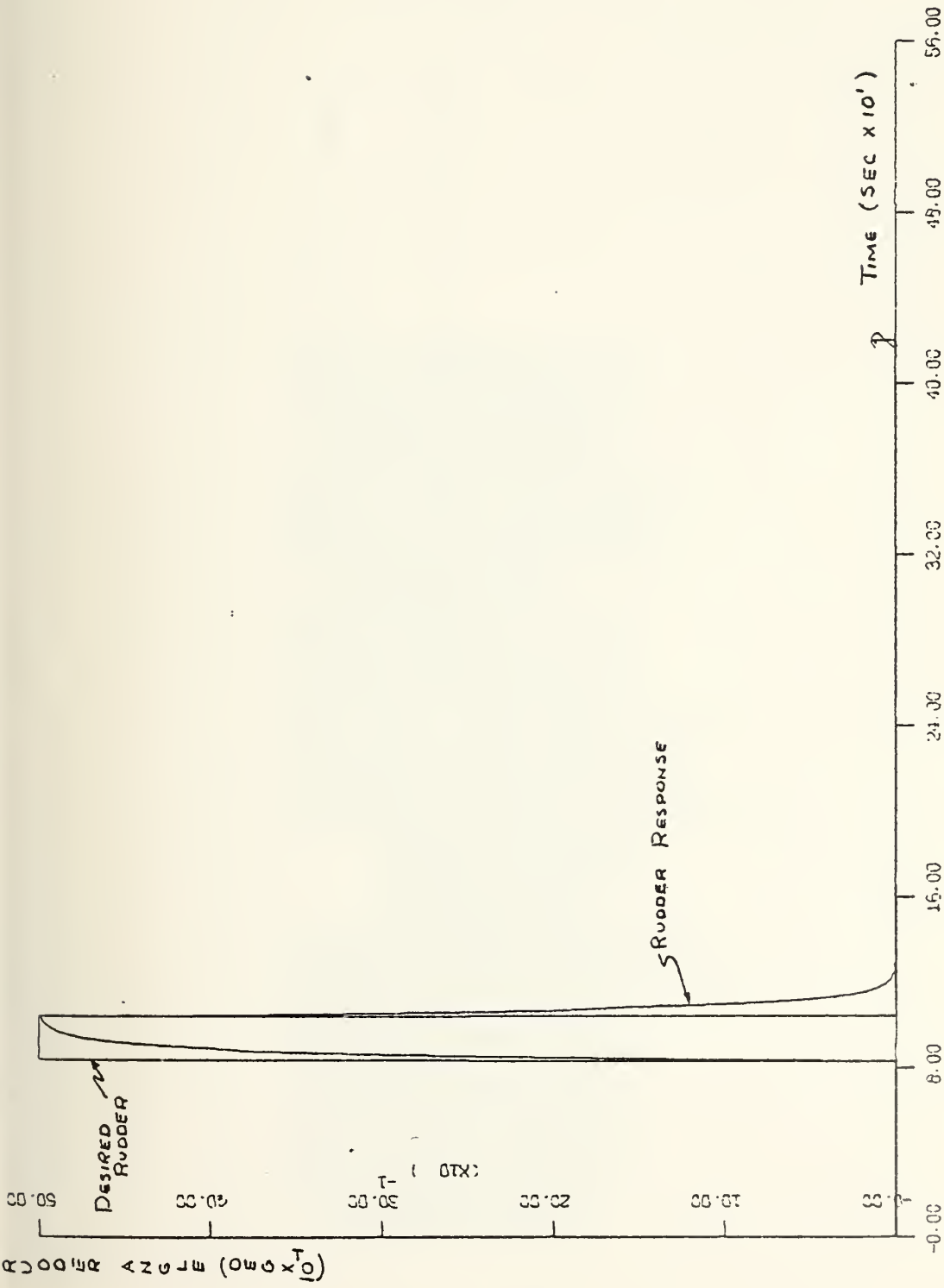


Figure III-13
Turn Phase Rudder Action of Reference Ship

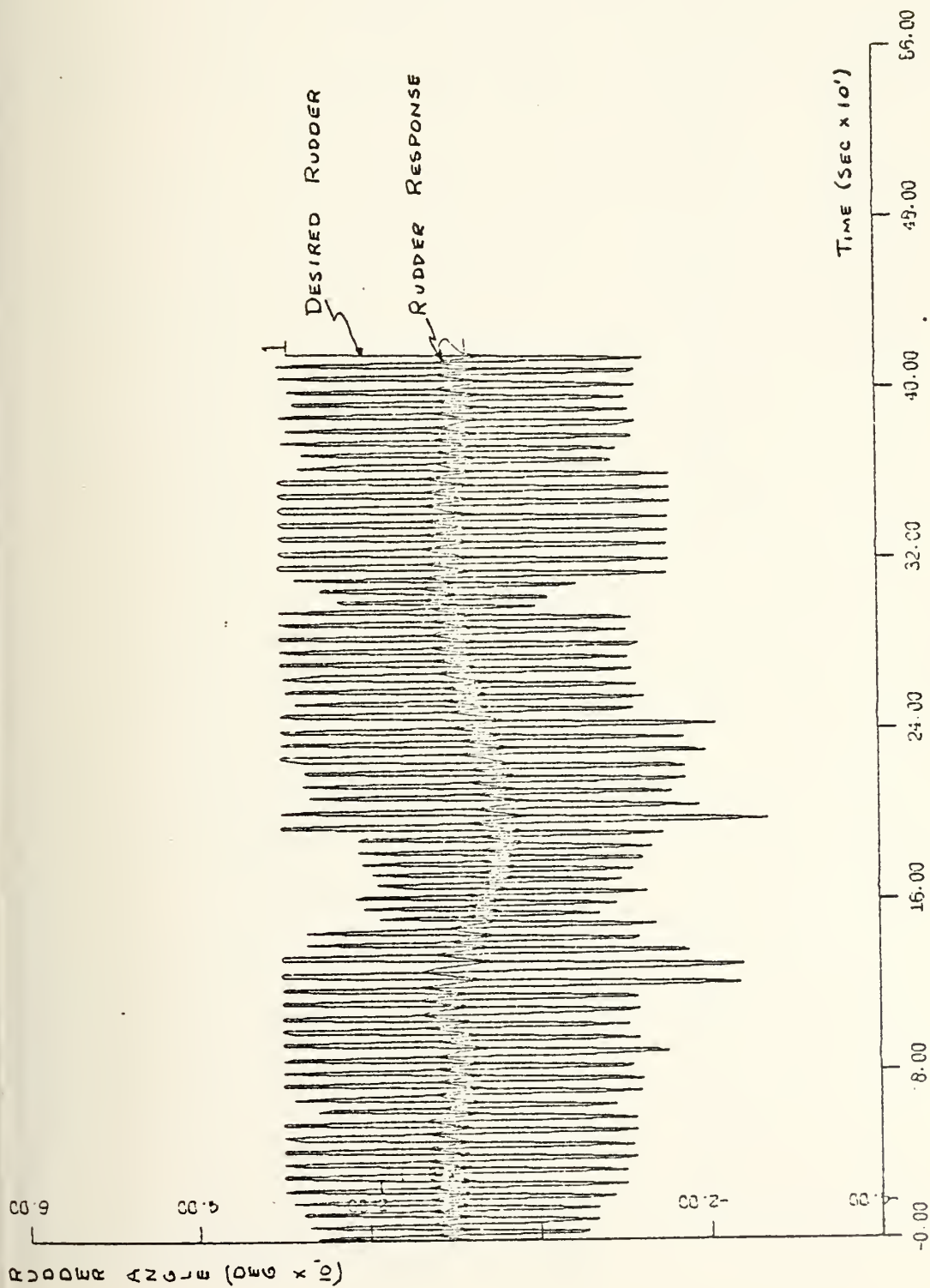


Figure III-14
Turn Phase Rudder Response



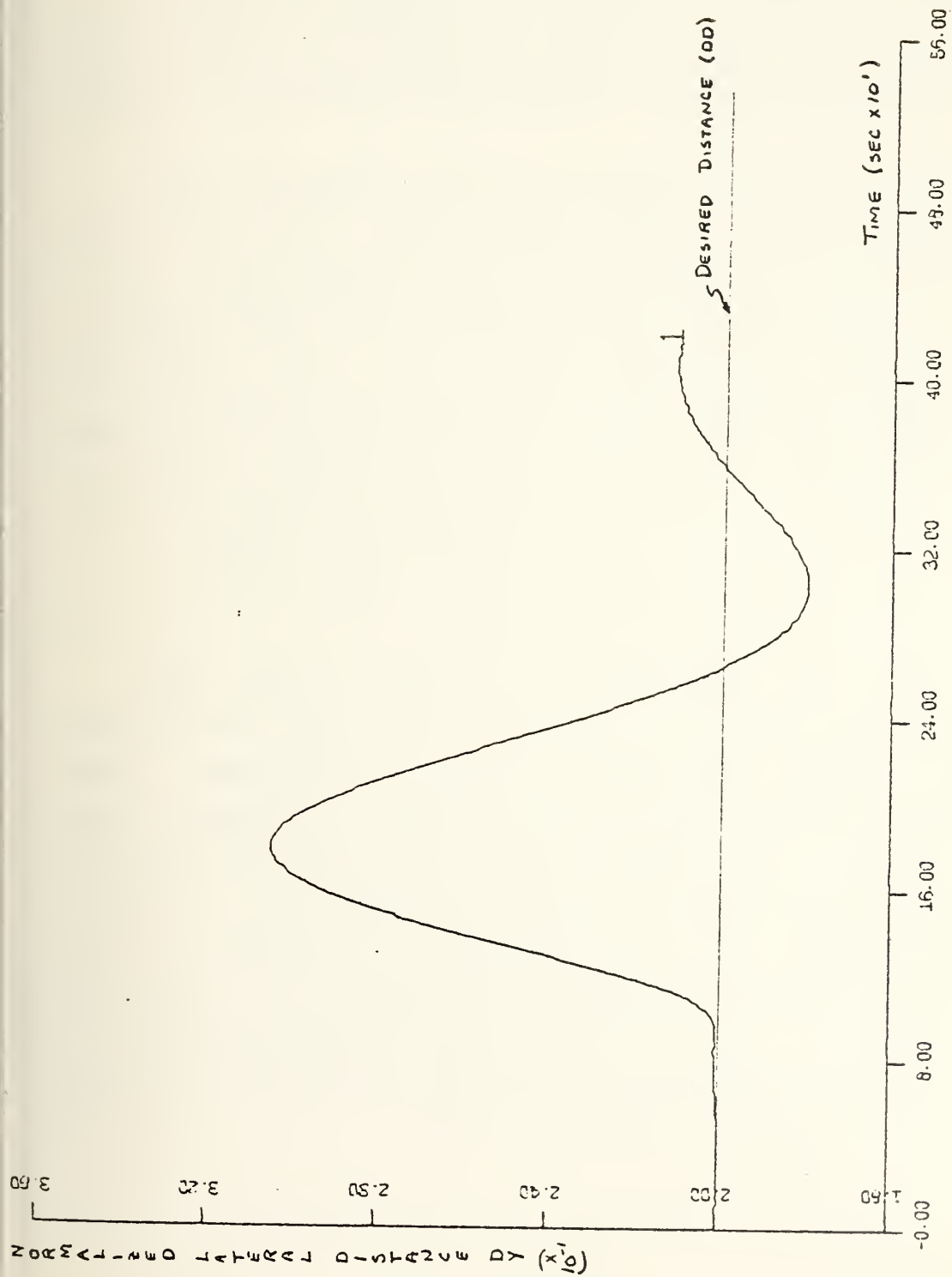


Figure III-15
Turn Phase Lateral Distance DY



Some initial perturbation is introduced by assuming the relative yaw angle when alongside is negligible.

e. second optimization

The same procedure was followed in obtaining gains that would optimize a chosen cost function. Figure III-6 still applies except that function FEA is replaced by function FEB (listed in appendix A) to simulate the new conditions.

Cost function criteria change in this instance since the ships start at the desired position and optimally stay at the same relative positions. Also, the rudder response to such a large turning perturbation must be free to cause achievement of the desired position. Due to these considerations, the integral of the absolute error (IAE) performance measure was chosen for the optimization criterion and can be written as:

$$OEJ = \int_{t_0}^{t_f} |ADY| dt$$

Table III-2 shows the results of the turn phase optimization and the comparison with the approach phase gains. Again DSL simulation was performed using the turn phase scenario. Figures III-16 thru III-21 portray the graphical results. The rudder response of figure III-21 indicates very sensitive response to the interactive forces shown in figures III-17 and III-18. The lateral distance separation of figure III-20 indicates excellent position maintainment with maximum excursion error of only 2 feet (0.0038 normalized). This minimal variation is well within that which can be tolerated in the RAS environment.



Gain	RSNS	WTSNS	RGN	VFBG
BU	2.0	20.0	50.0	10.0
BL	0.1	0.1	1.0	0.01
XS	1.0	1.0	10.0	1.0
OUTPUT	1.99765	0.7357	49.9776	0.084028
OBJ	0.009145			
Approach Phase Output	1.86642	2.3869	23.4185	4.35162

Table III-2
Turn Phase Optimization Results



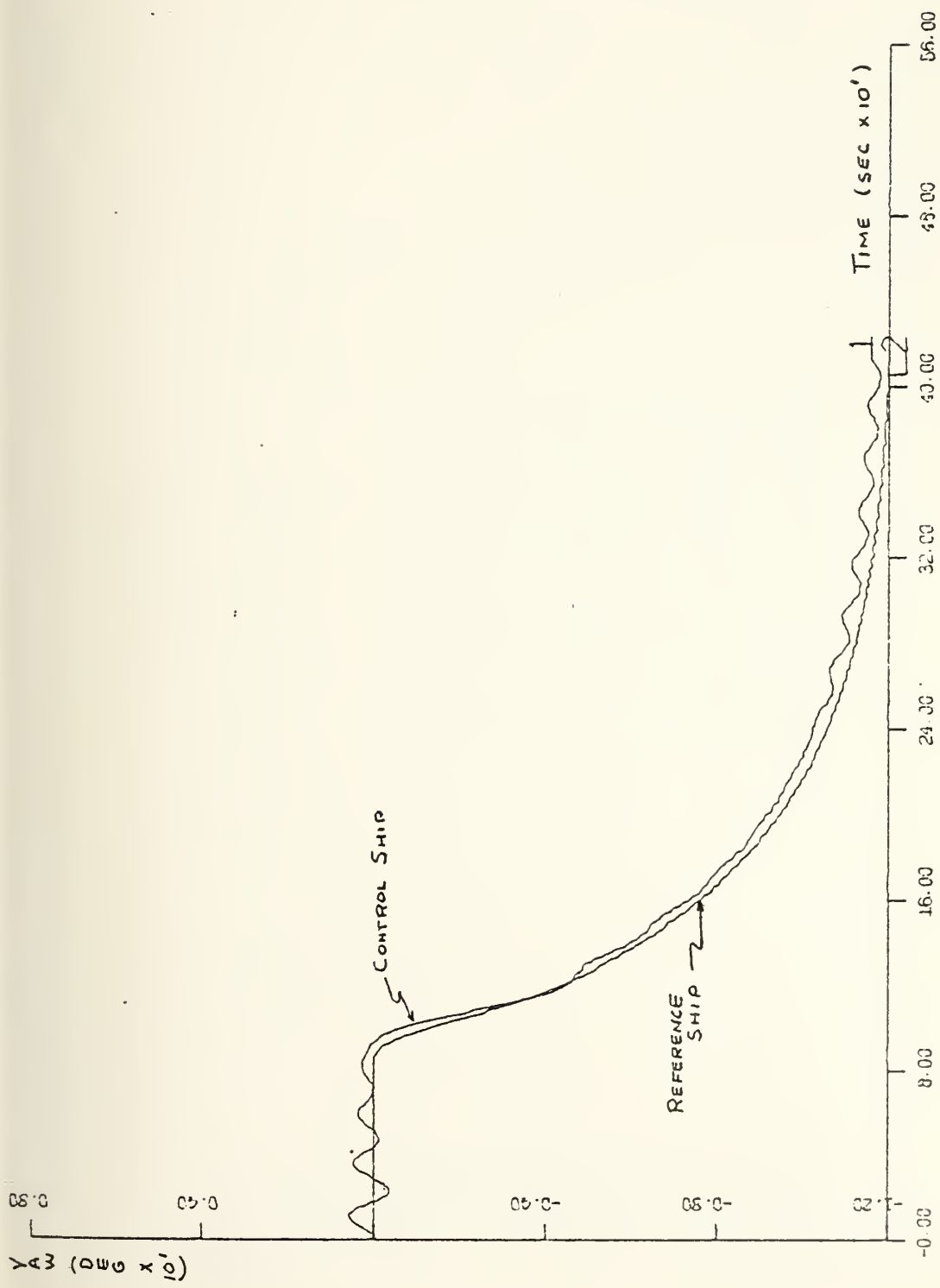


Figure III-16
Turn Phase Yaw Response



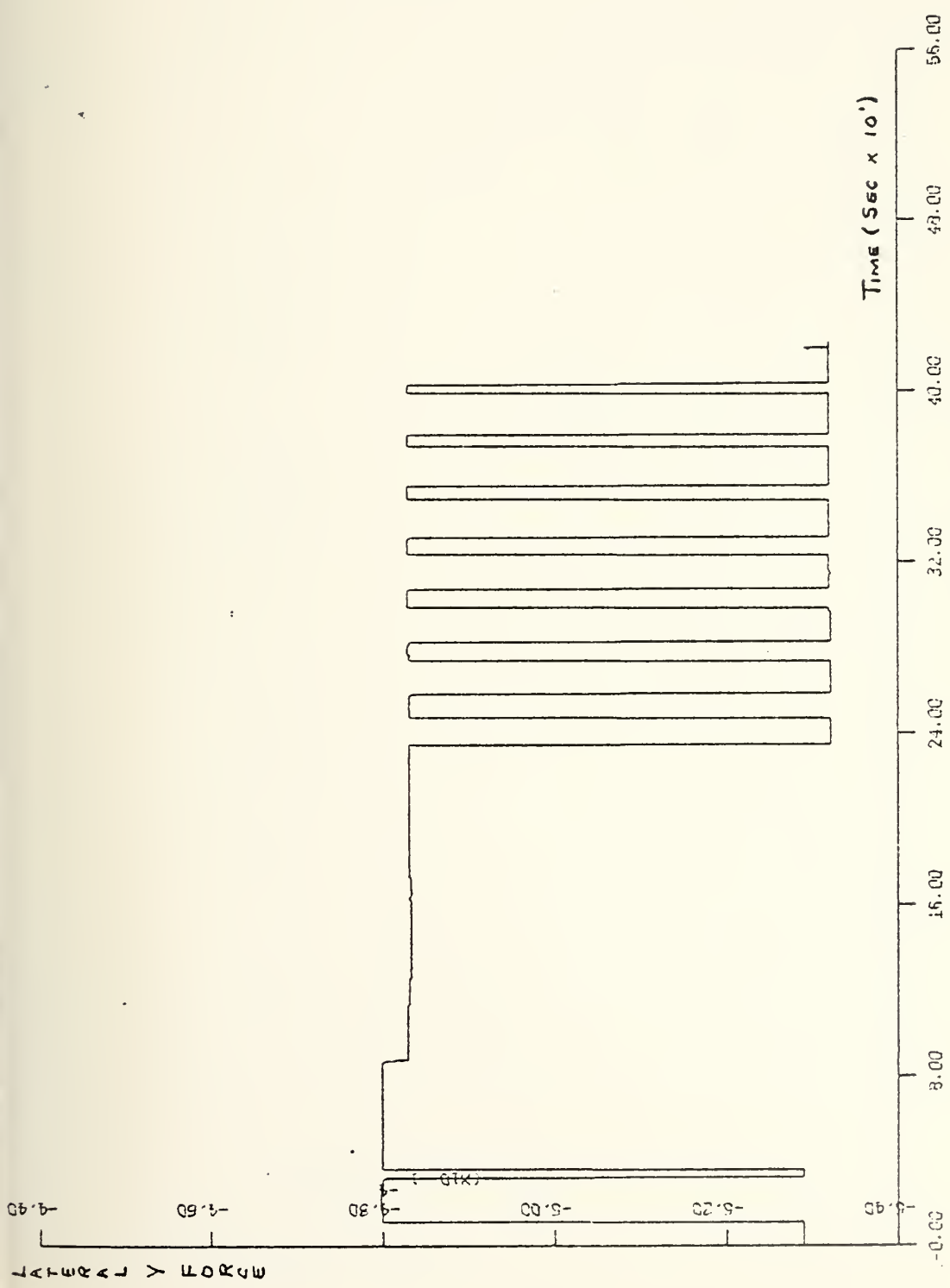


Figure III-17
Turn Phase Y Forces

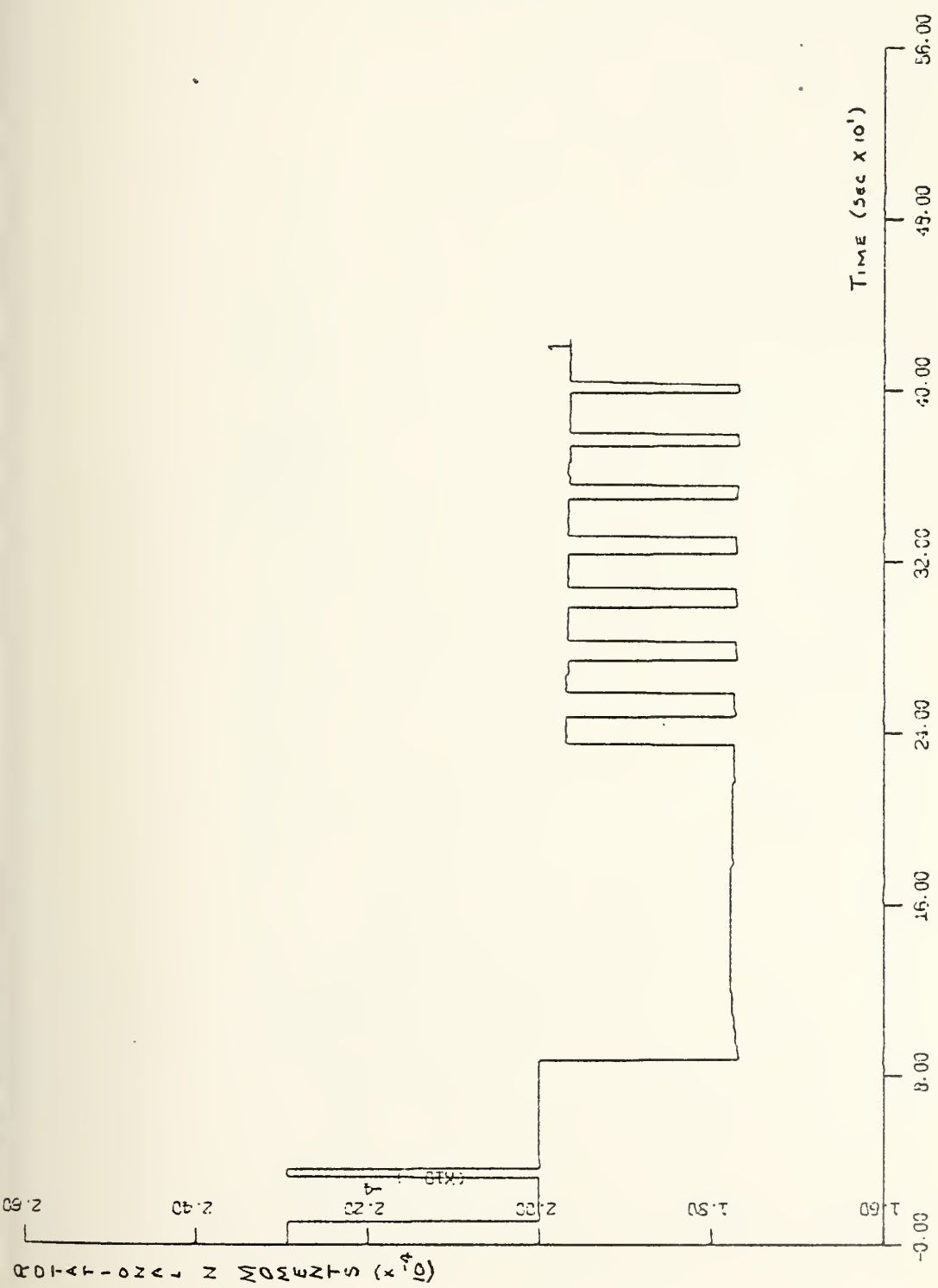


Figure III-18
Turn Phase N Moments



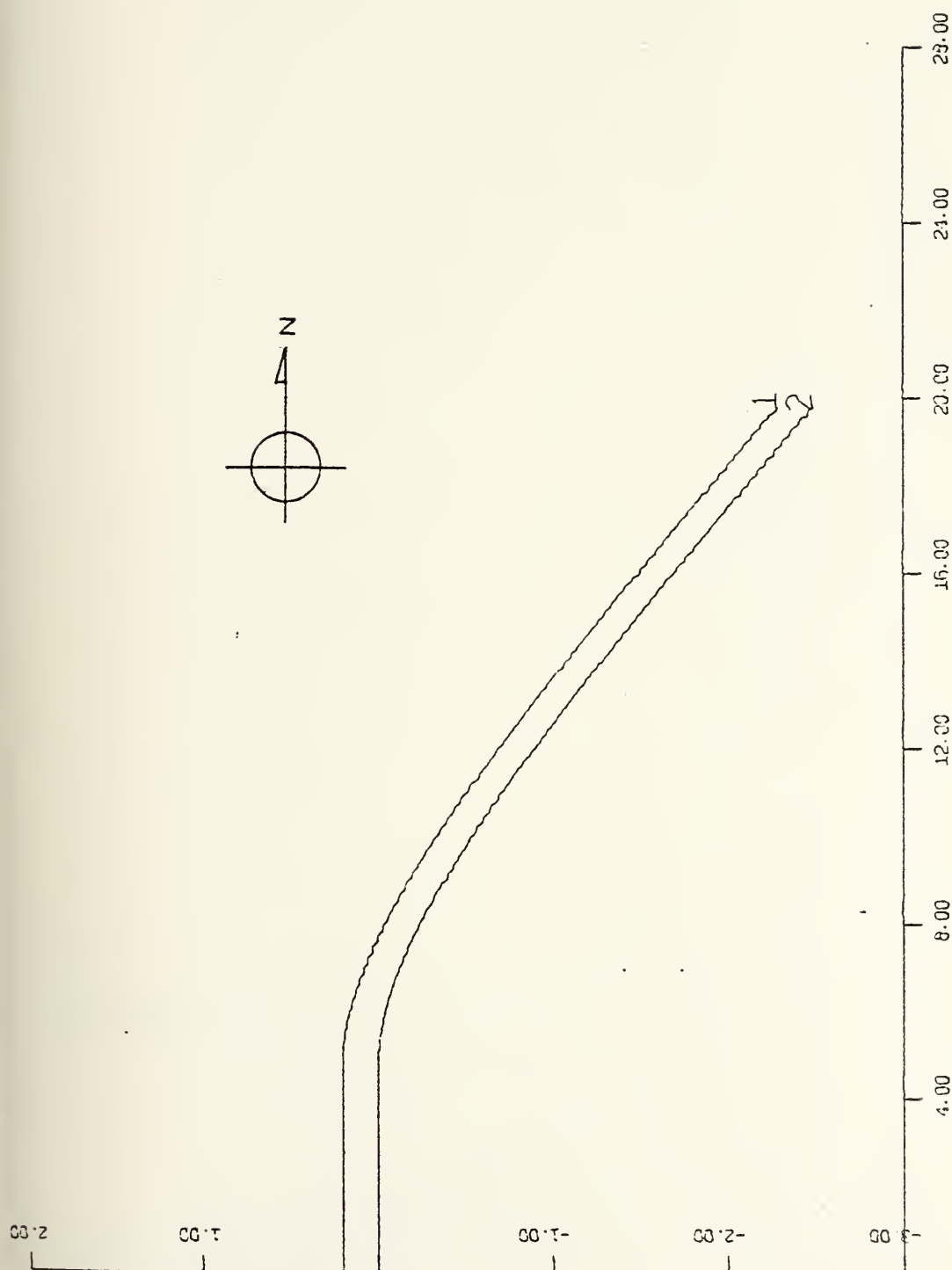


Figure III-19
Turn Phase Geographic Plot

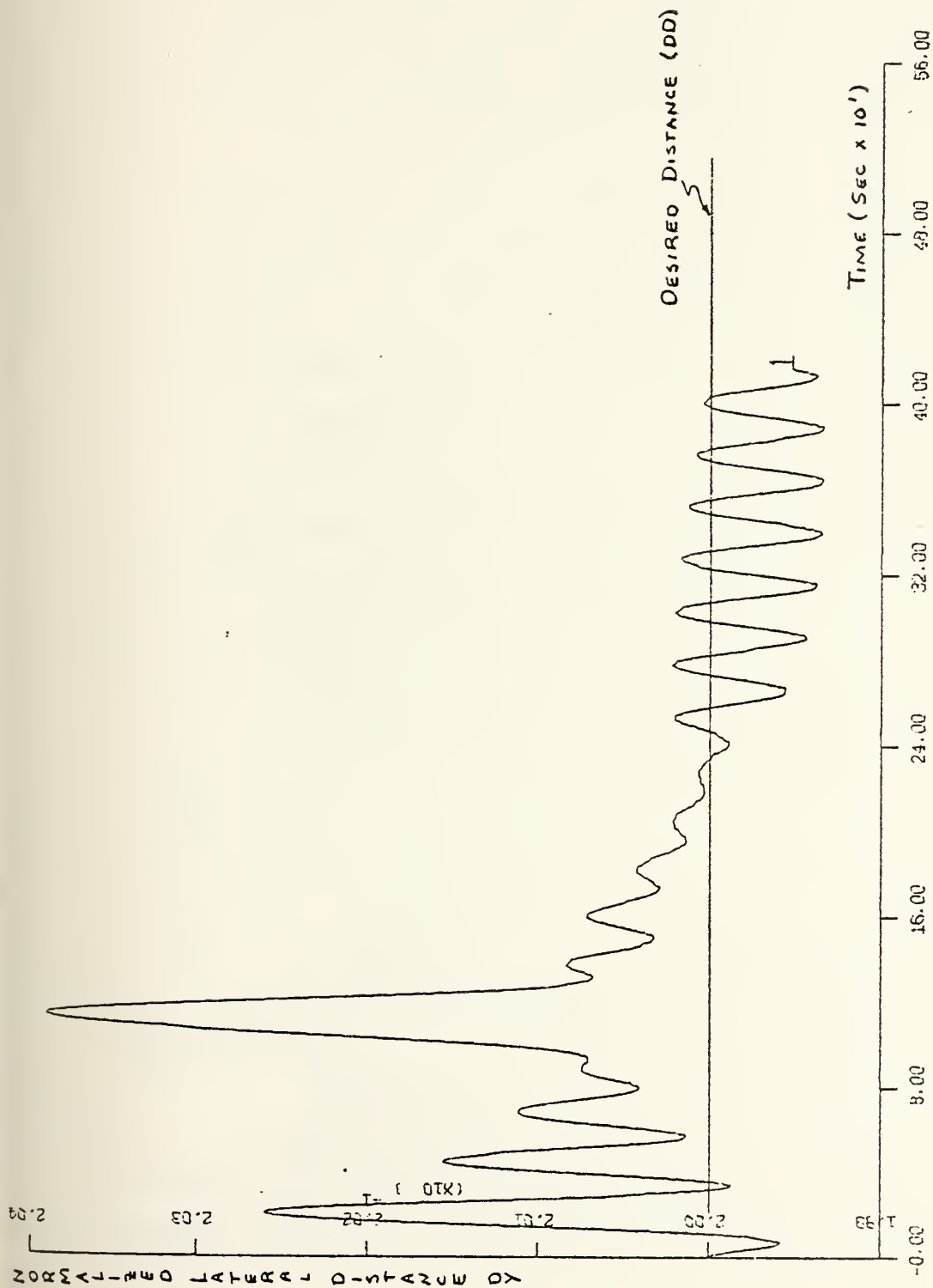


Figure III-20
Turn Phase Lateral Distance DY

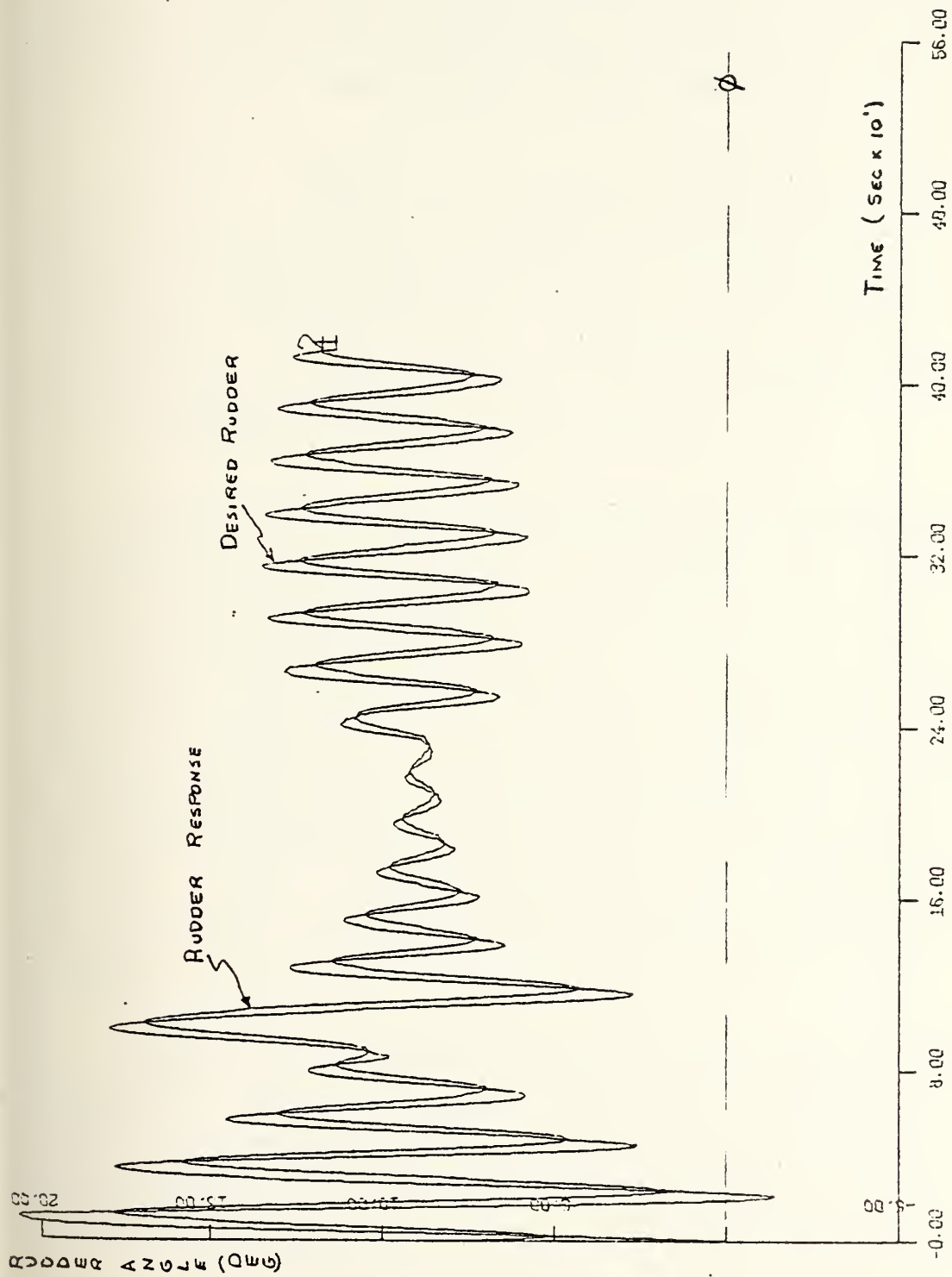


Figure III-21
Turn Phase Rudder Response



f. Continued Control Testing

To alleviate suspicions that the response from the gains obtained in the approach phase could be improved by the gains obtained in the turn phase, a simulation of the approach phase was accomplished with the new gains. Figure III-22 is the graphic display of the effect of these gains on the approach phase lateral distance positioning.

Careful analysis of the results thus far clearly indicate the need for an adaptive control scheme to allow gain adaptation to meet the design specifications. A full adaptive control scheme for systems of this type is outside the scope of this thesis. References 15 thru 23 are indications of some of the literature available for pursuit of a completely adaptive control system.

What was done here is development of a simple algorithm to sense when the conditions were adequate to switch from one set of gains to another. This may be done with the two sets of gains developed thus far. However, for the sake of simulation efficiency, a third set of gains was introduced. This third set amounts to a change of one approach gain (RSENS) which has previously been defined as the range sensitivity gain. The simulation efficiency is increased by decreasing the time required for the approach phase to reach steady state. A consequence of this procedure is a reenforcement of the need for a completely adaptive control scheme.

Repeated simulation revealed that commencing the turn (in effect switching gains), before a reasonable steady state was reached caused results similar to those shown in figure III-22. An increase in RSENS to a value of 4.0 when the lateral separation error is less than 0.05 (normalized)



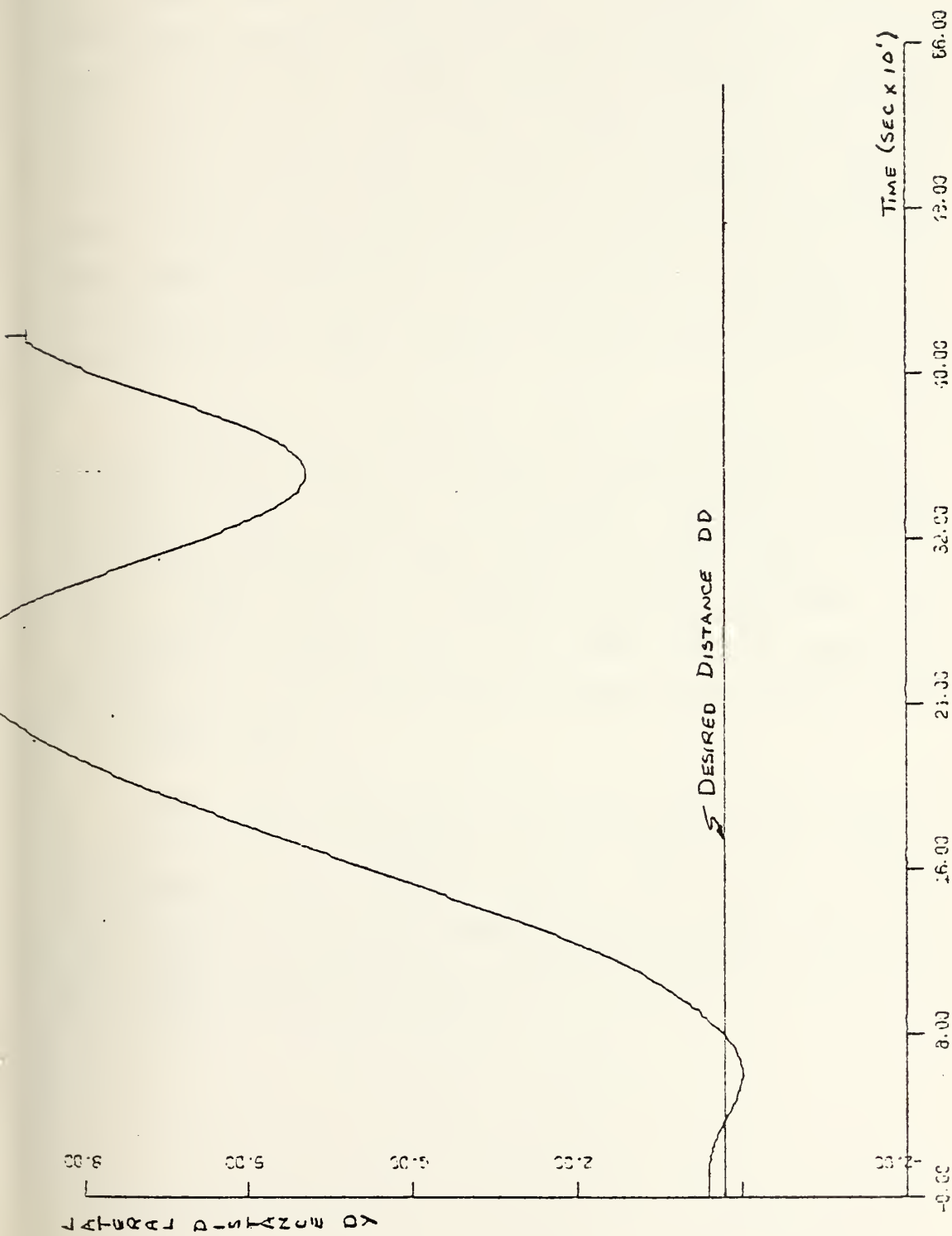


Figure III-22
Approach Phase Lateral Distance DY

and greater than 0.005 (normalized) forces acceptable steady state in approximately 1/2 the time previously required using a single set of approach gains.

Subroutine SWITCH (listed in appendix A) incorporates this simple adaptive gain schedule with a counter mechanism to sense when steady state is reached. Further study indicated a need to damp the yaw oscillations to a greater extent if the yaw velocity (BDOT2D) exceeded 2.0 degrees/sec when the gains are initially switched to the more sensitive ones of the turn phase. This is an artificial adaptive gain for VFEG caused by computer time restrictions prevalent in a full scale computer simulation where both the approach and turn phases are desired. If the gain switching point is moved up in time, as would normally be the case in a real life situation, this damping increase would not be required.

The results of the full scale simulation using computer program #6 are shown in figures III-23 thru III-34. The approach phase plots of figures III-23 thru III-28 show definite improvement over that previously shown in figures III-7 thru III-12. Figure III-27 indicates that the overshoot is reduced to 10.6 feet (0.02 normalized) as opposed to 17.9 feet (0.034 normalized) that was prevalent in figure III-11.

The turn phase plots are shown in figures III-29 thru III-34 and show responses very similar to those shown previously in figures III-16 thru III-21. The only significant differences occur in the initial responses which are due to the incorrect initialization when the turn phase was simulated individually.

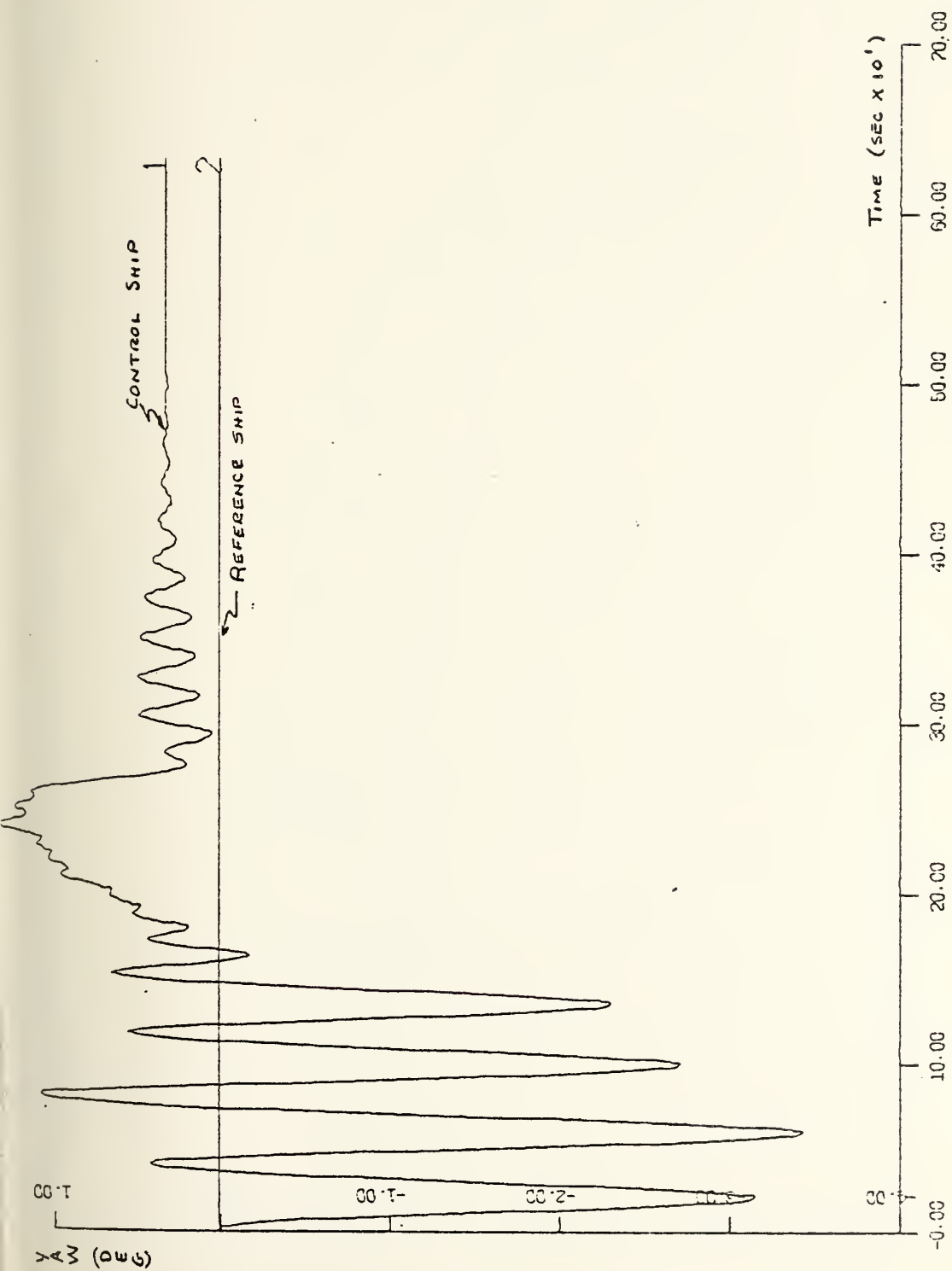


Figure III-23
Approach Phase Yaw Response

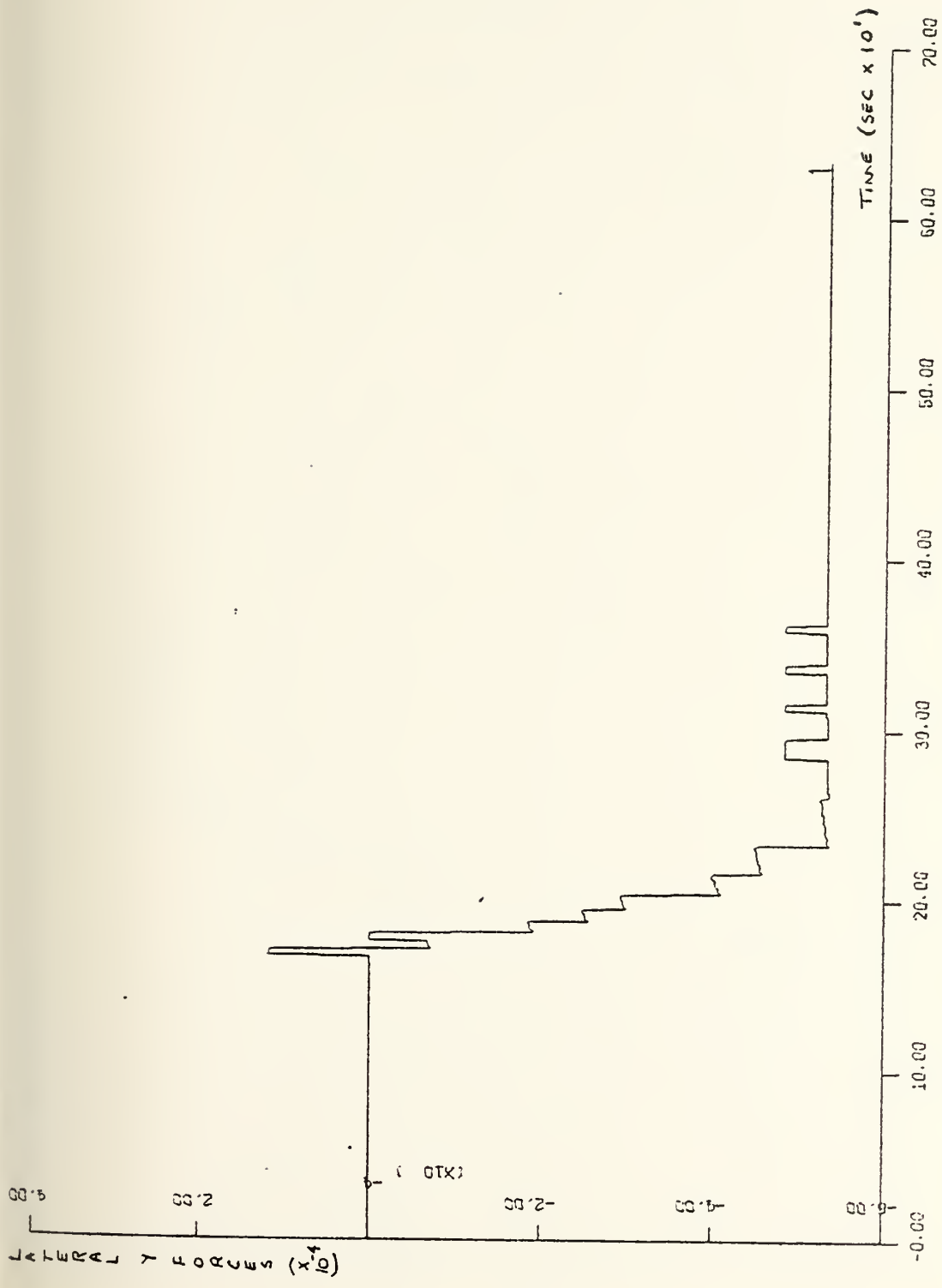


Figure III-24
Approach Phase Y Forces

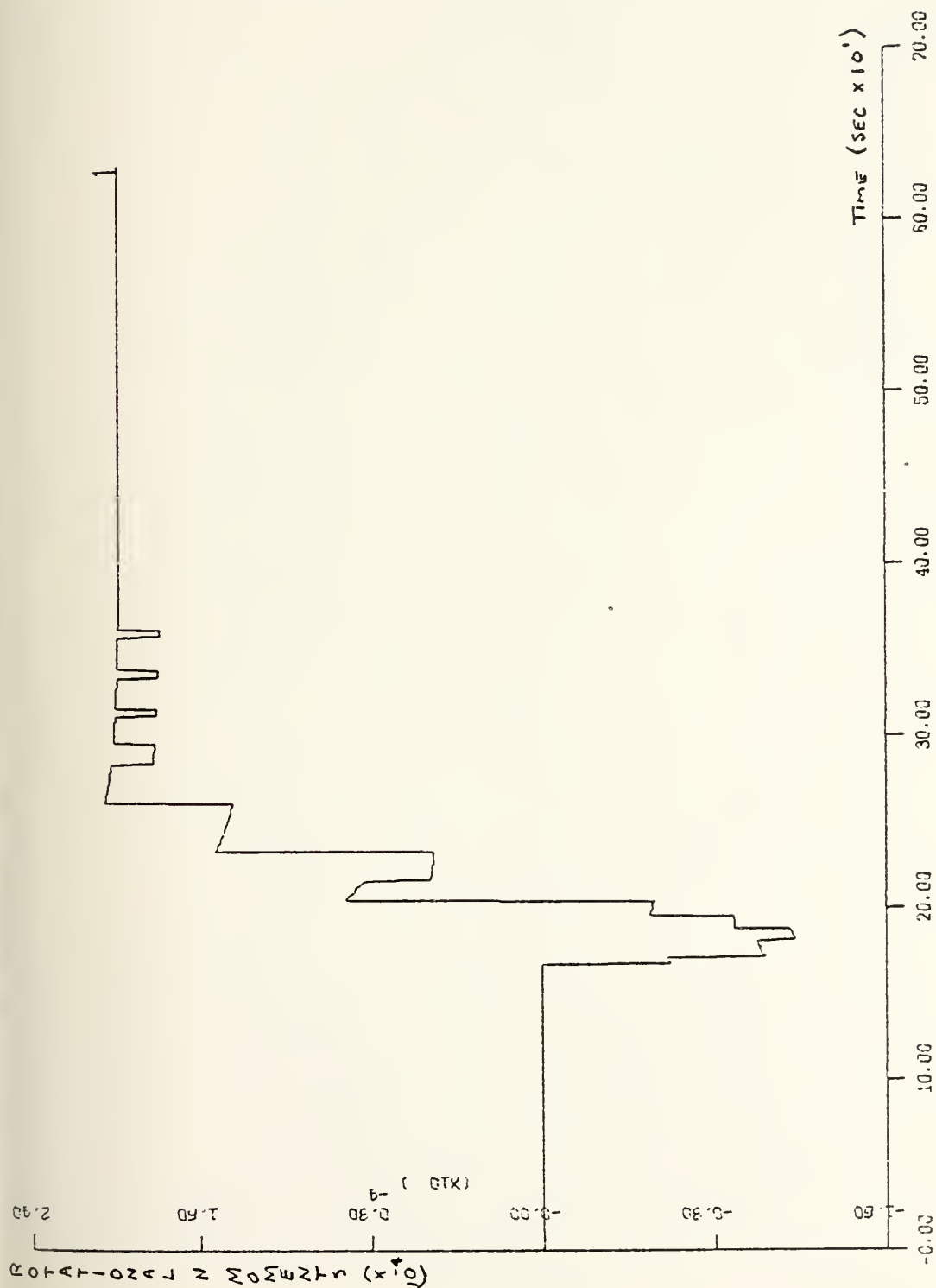


Figure III-25
Approach Phase N Moments

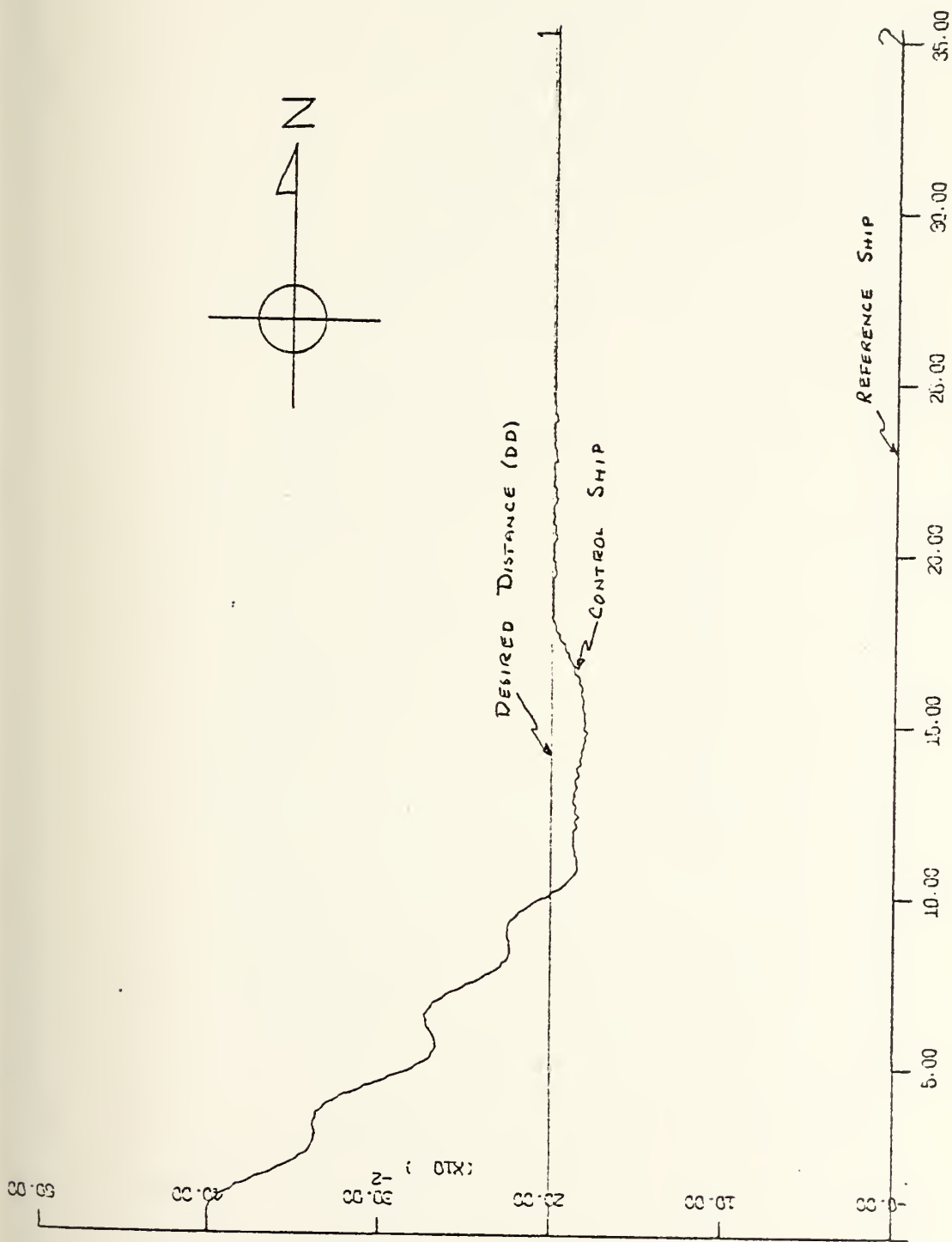


Figure III-26
Approach Phase Geographic Plot

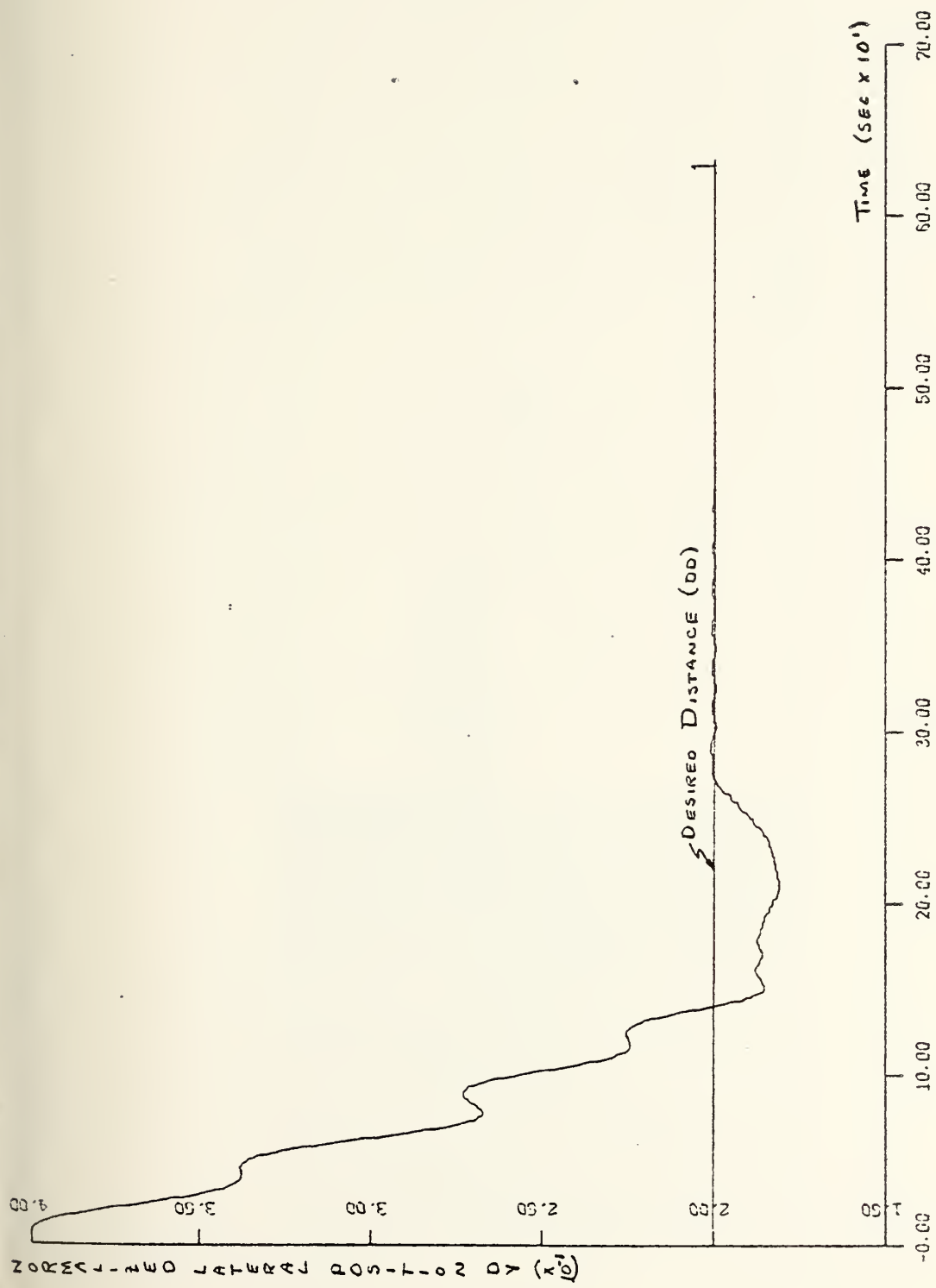


Figure III-27
Approach Phase Lateral Distance DY

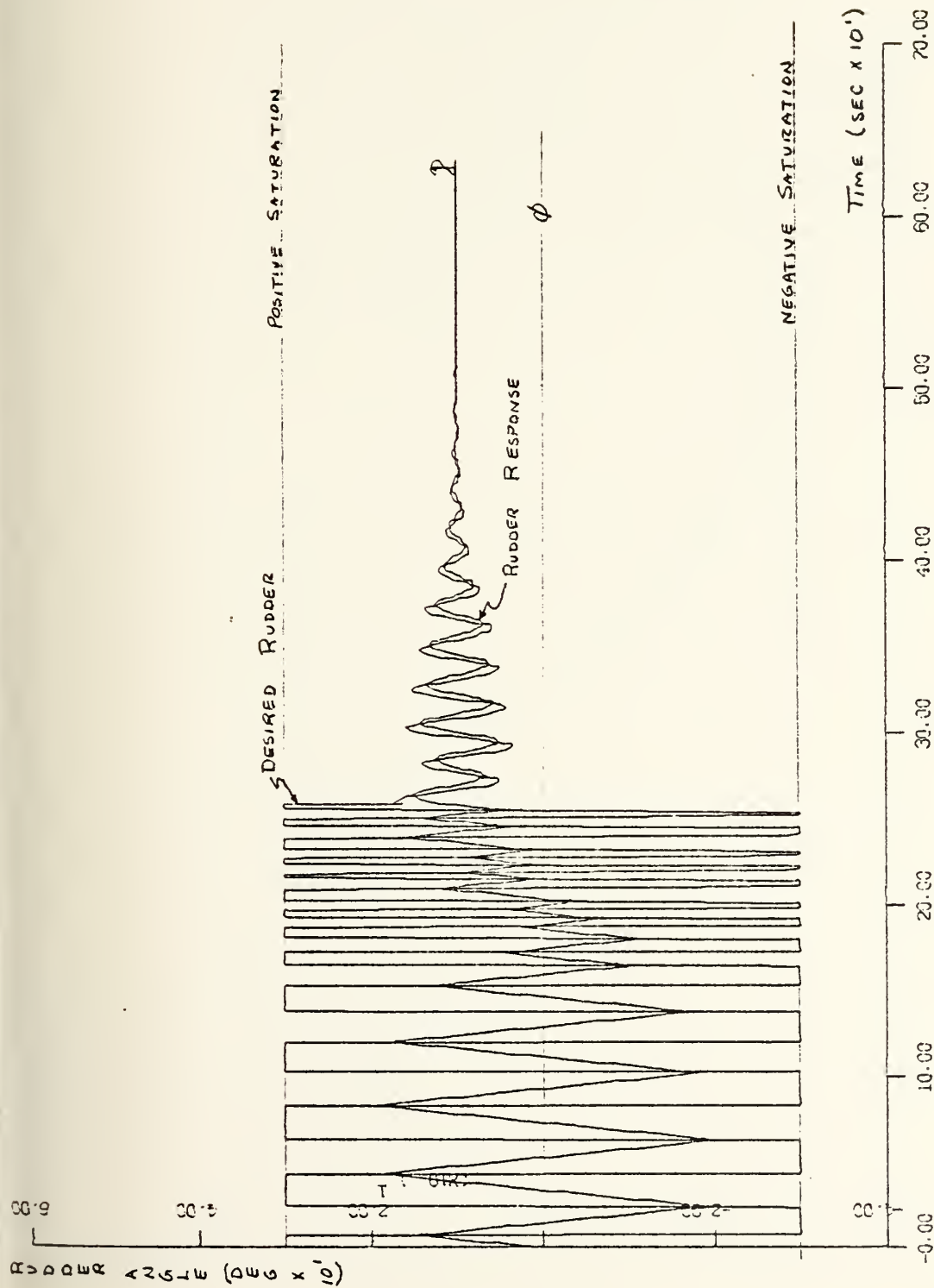


Figure III-28
Approach Phase Rudder Response



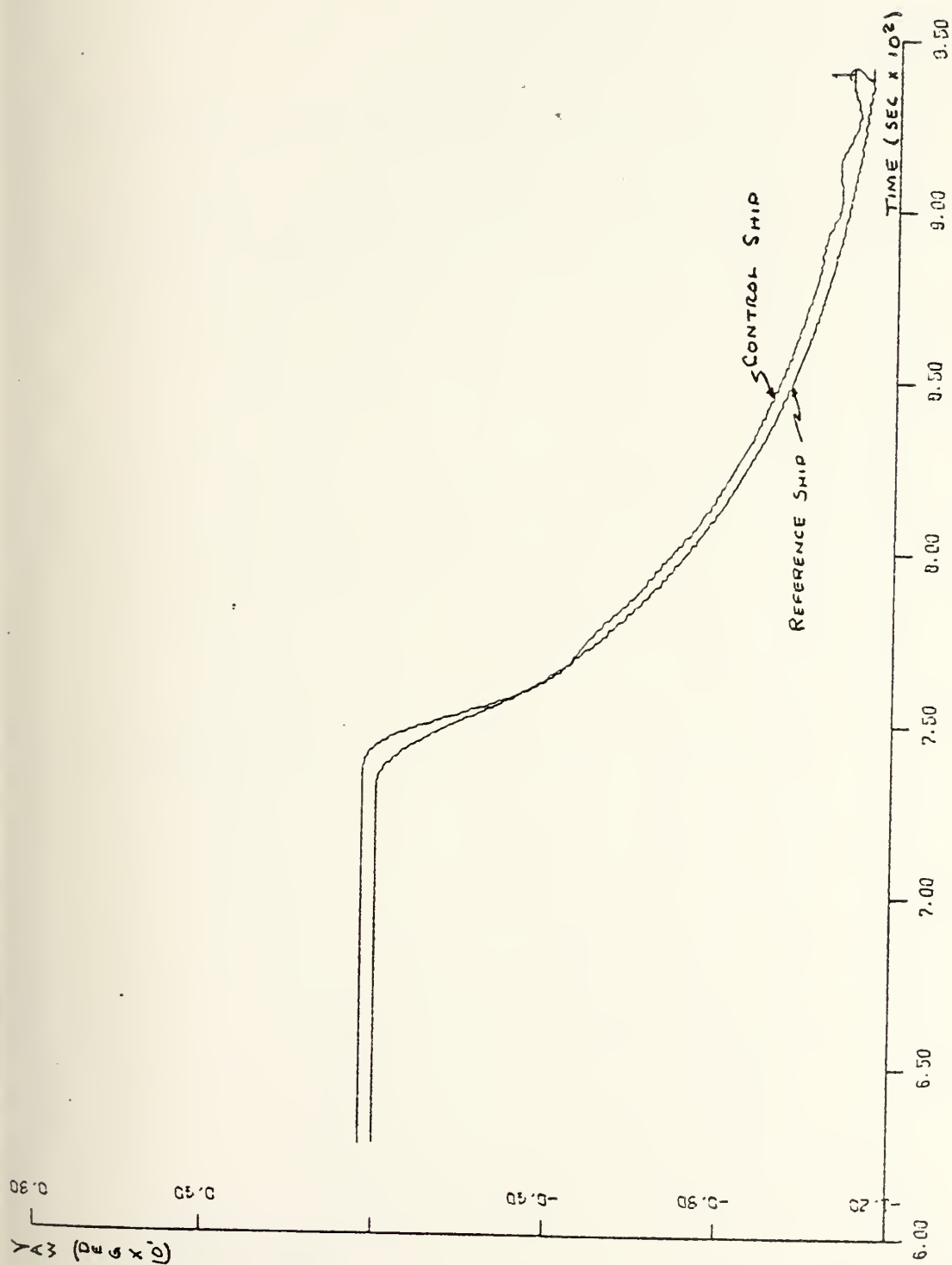


Figure III-29
Turn Phase Yaw Response



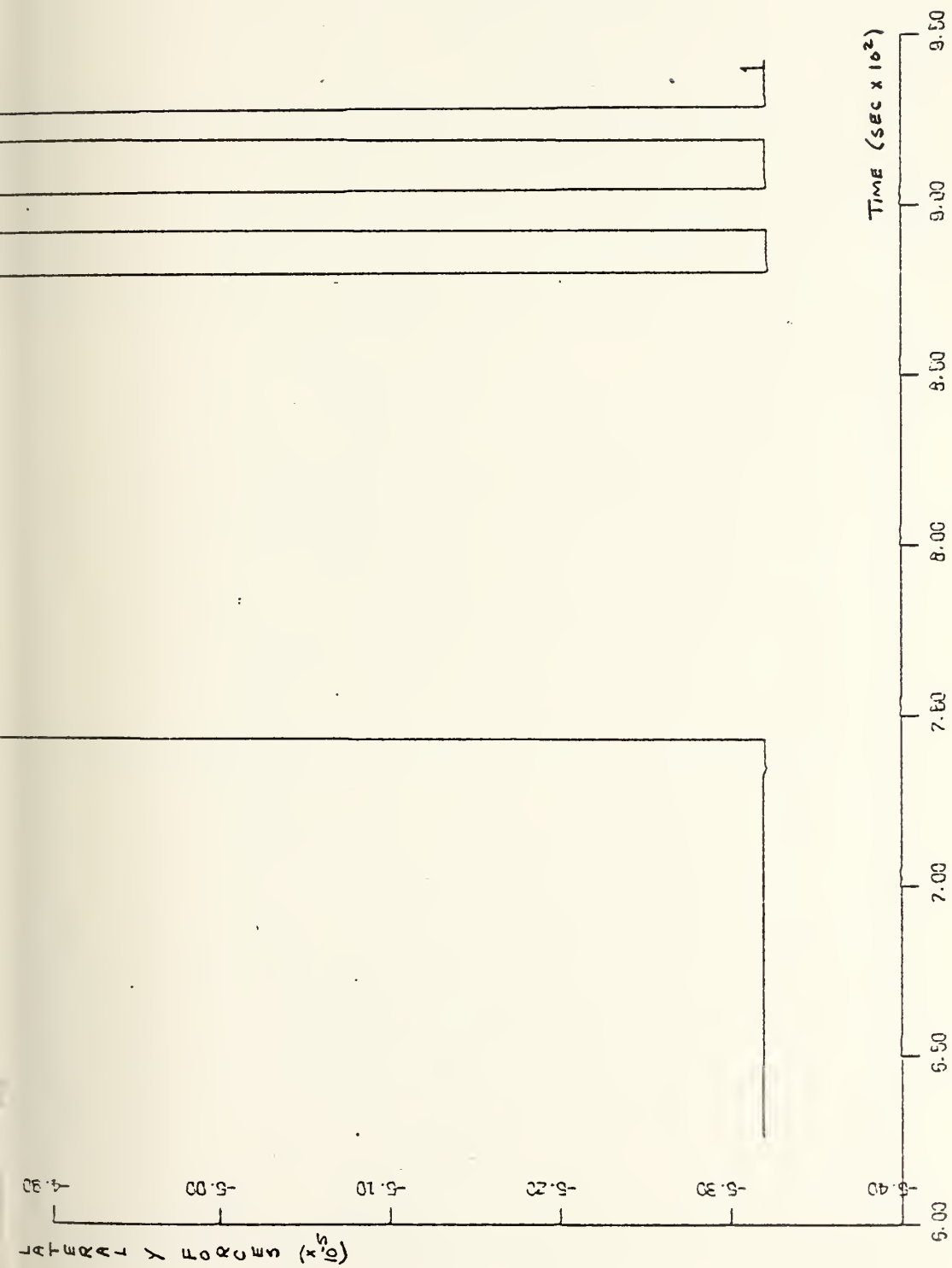


Figure III-30
Turn Phase Y Forces

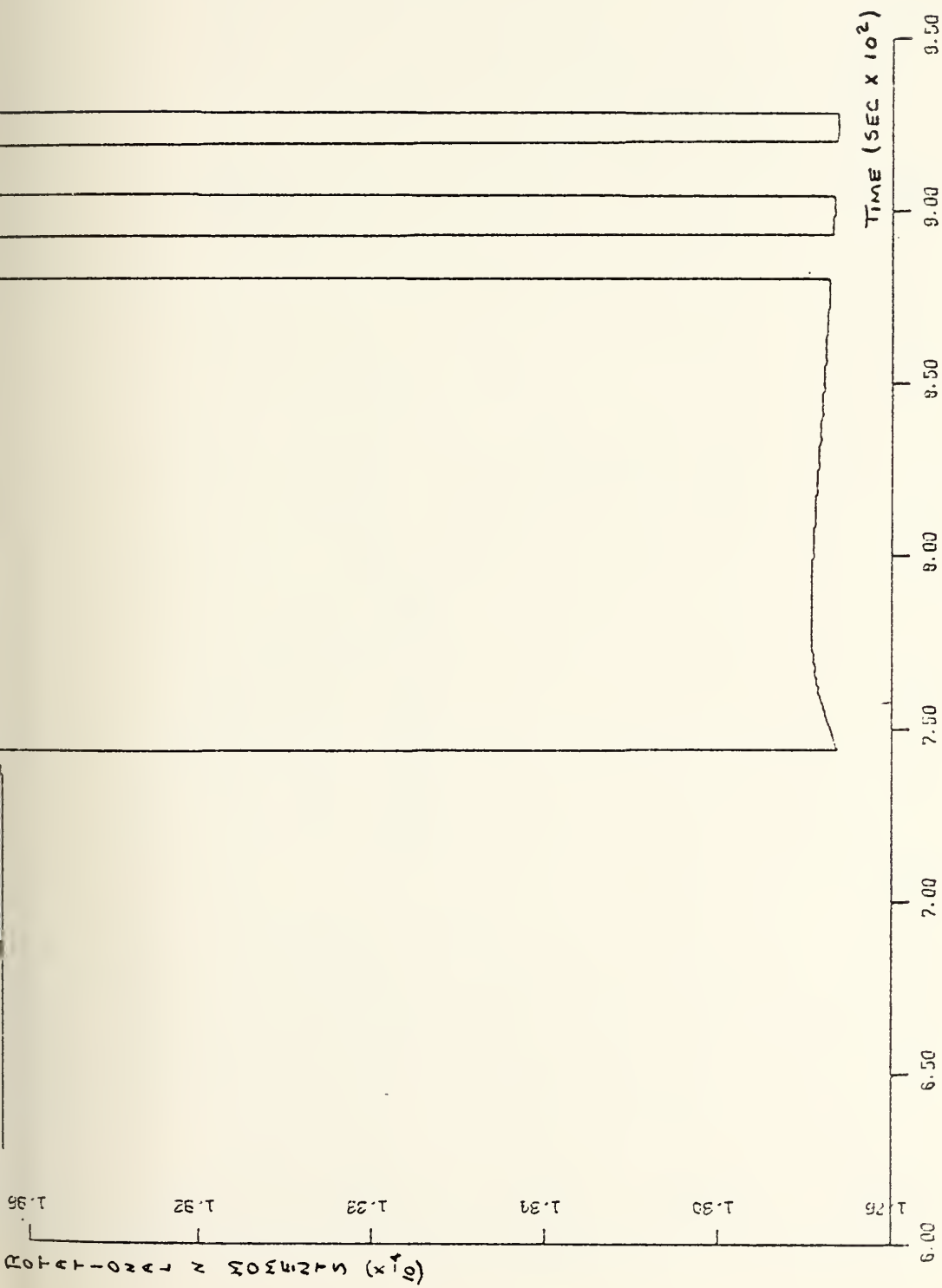


Figure III-31
Turn Phase N Moments

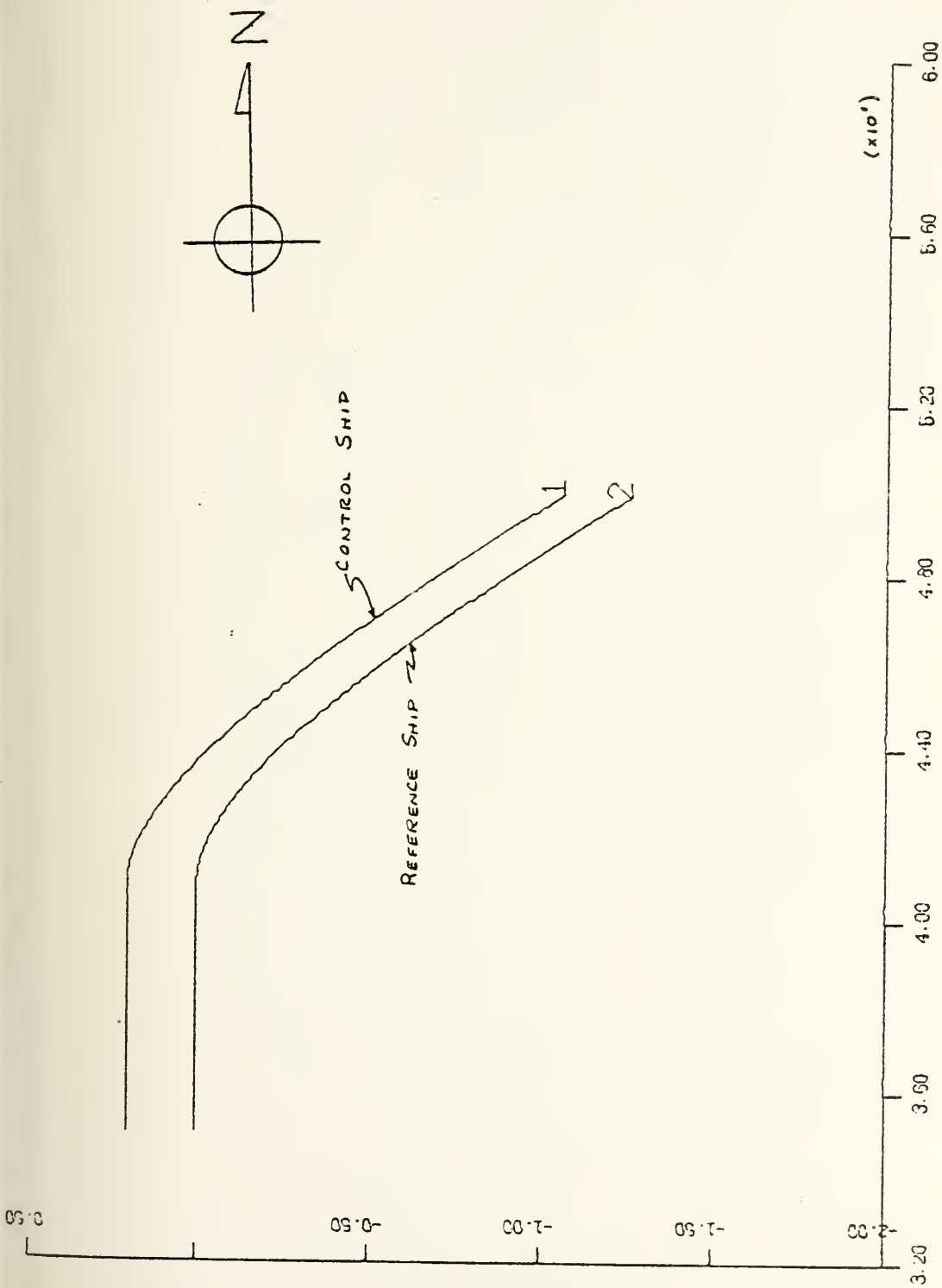


Figure III-32
Turn Phase Geographic Plot

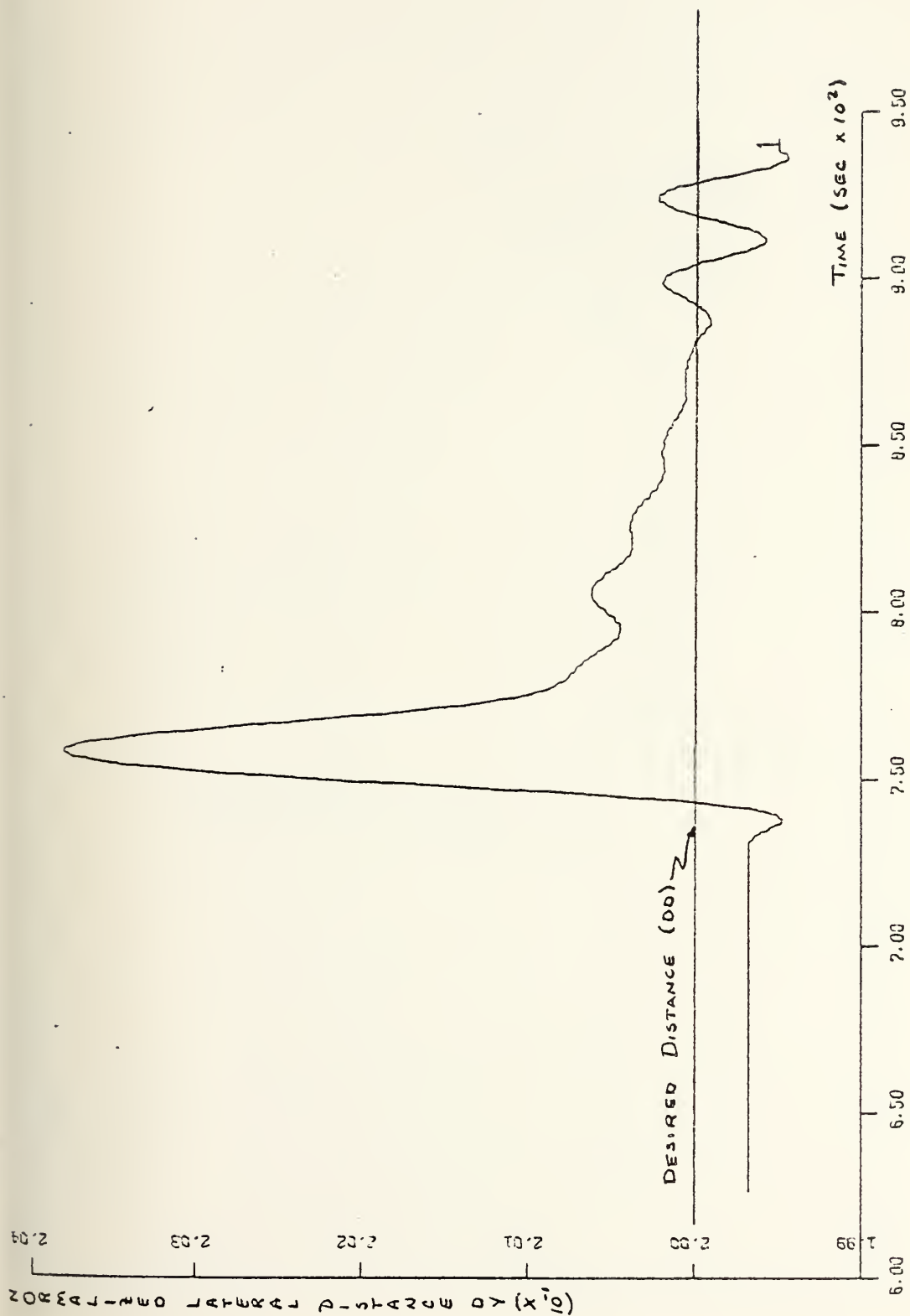


Figure III-33
Turn Phase Lateral Distance DY

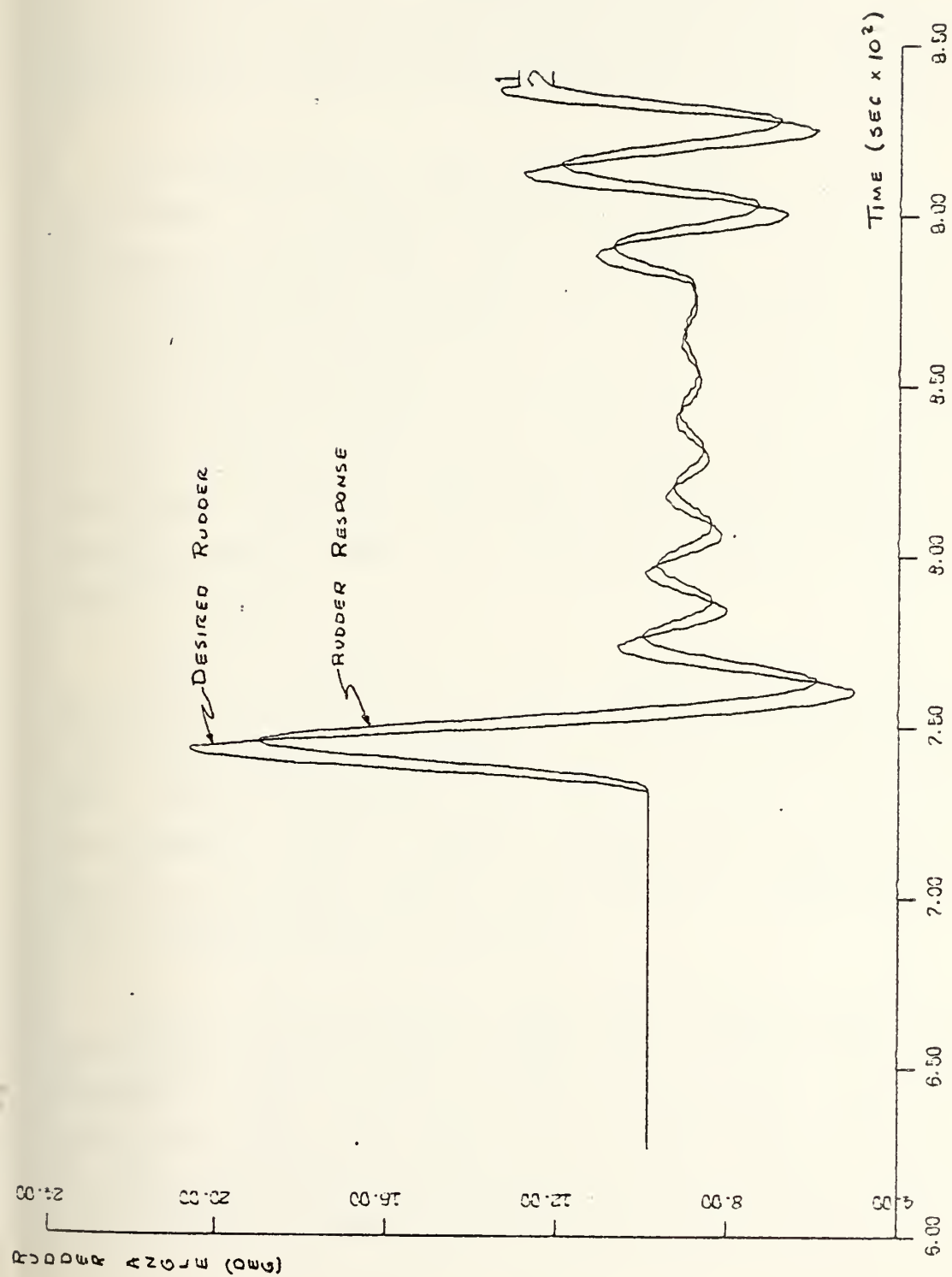


Figure III-34
Turn Phase Rudder Response

g. Varying Initial Conditions

The results obtained in the previous section are most gratifying but actually incomplete. This system must work for other initial conditions quite different from those envisioned in the optimization scenario. The initial approach can realistically commence at points other than 5 ship lengths astern and displaced by 0.4 ship lengths.

By simulating this system with varying initial positions, the relative efficiency and worth of the control system can be observed. This was done in successive test runs whose initial conditions and corresponding plot figures are tabulated in table III-3. For the sake of brevity only those figures required to illustrate the relative efficiency of the control system are included. The corresponding initial optimization simulation figures are listed for cross reference. The turn phase plots for all runs except 4 and 6 exactly match that of the initial simulation and are not repeated here.

Runs 3, 5 and 6 were accomplished to show that no ambiguities exist in the control scheme to prohibit adequate real life initial conditions. Run 3 simulates the situation most often encountered by this author in the RAS environment. This scenario starts the control ship dead astern at 5.0 ship lengths and brings it alongside at 0.2 ship lengths lateral separation.

Run 5 is a situation where the approaching ship is purposely placed out of position on the wrong side for approach. The control scheme adequately corrects the placement error and will do so for all cases of this type, provided that there is adequate maneuvering room astern of the reference vessel (in this case 2.6 ship lengths was

	RUN	Initial Development	1	2	3	4	5	6
Initial Condition	X01	5.0	4.0	3.0	5.0	5.0	5.0	5.0
	Y02	0.4	0.3	0.25	0.0	0.2	-0.4	-0.4
	DD	0.2	0.2	0.2	0.2	0.15	0.2	0.2
Desired Distance	IS	STBD	STBD	STBD	STBD	STBD	STBD	PORT
Approach Side								
Approach Phase Figures (III-)								
YAW		23	35	38	41	44	47	50
Geographic Plot		26	36	39	42	45	48	51
Rudder Response		28	37	40	43	46	49	52
Turn Phase Figures (III-)								
YAW			29			53	29	56
Geographic Plot			32			54	32	57
Rudder Response			34			55	34	58

Table III-3
Initial Condition Simulation Cross Reference



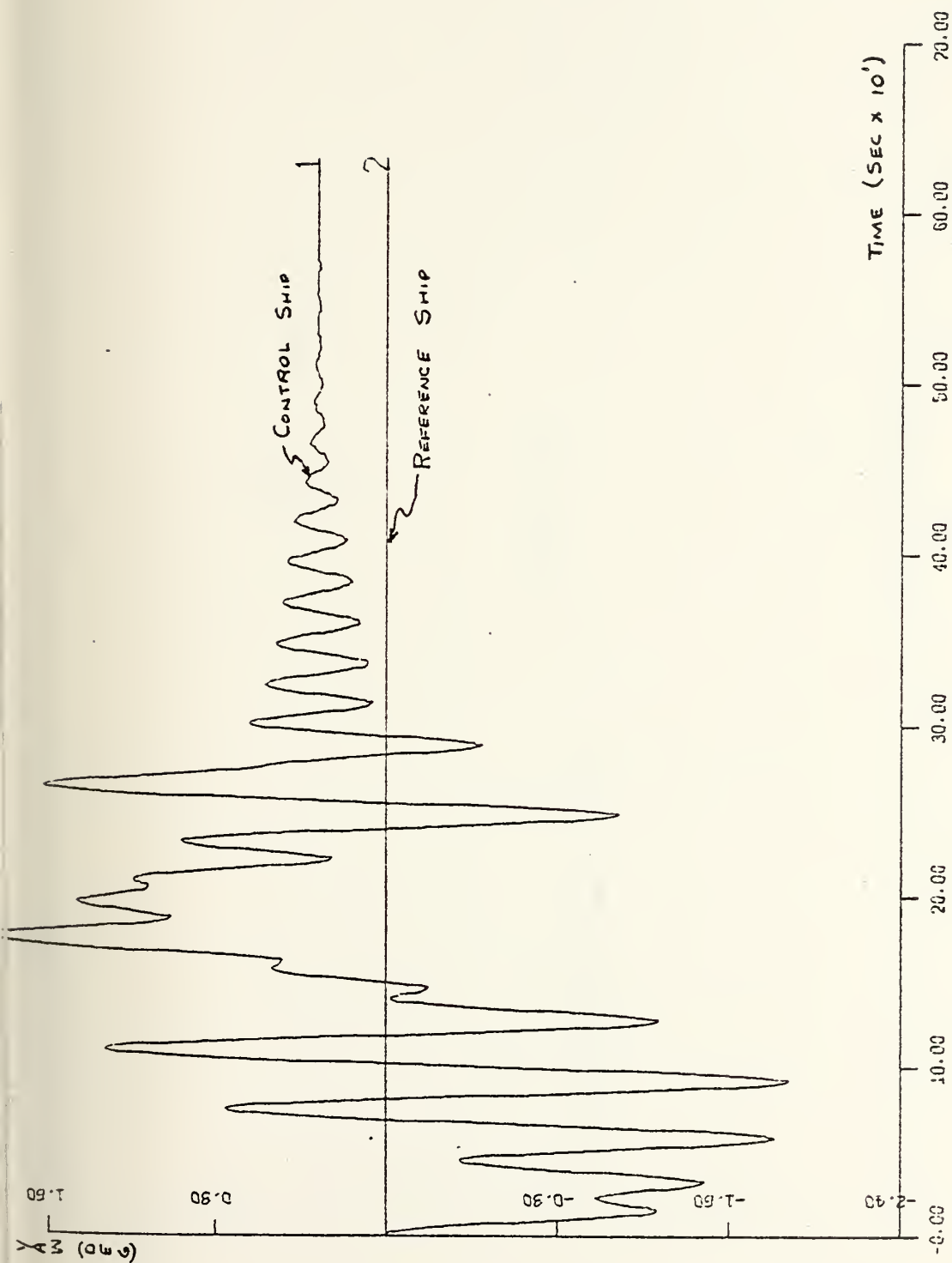


Figure III-35
Approach Phase Run #1 Yaw Response

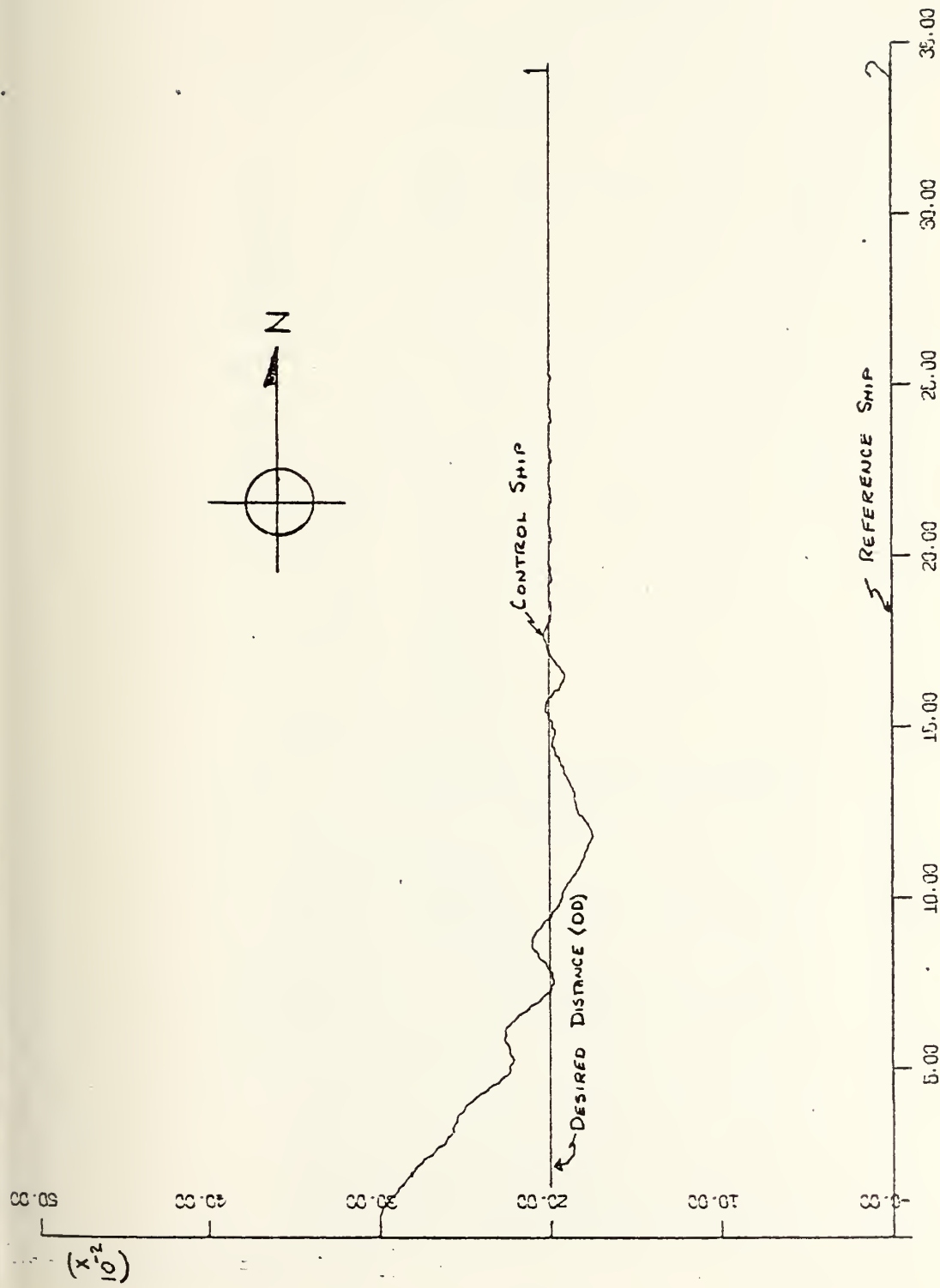


Figure III-36
Approach Phase Run #1 Geographic Plot

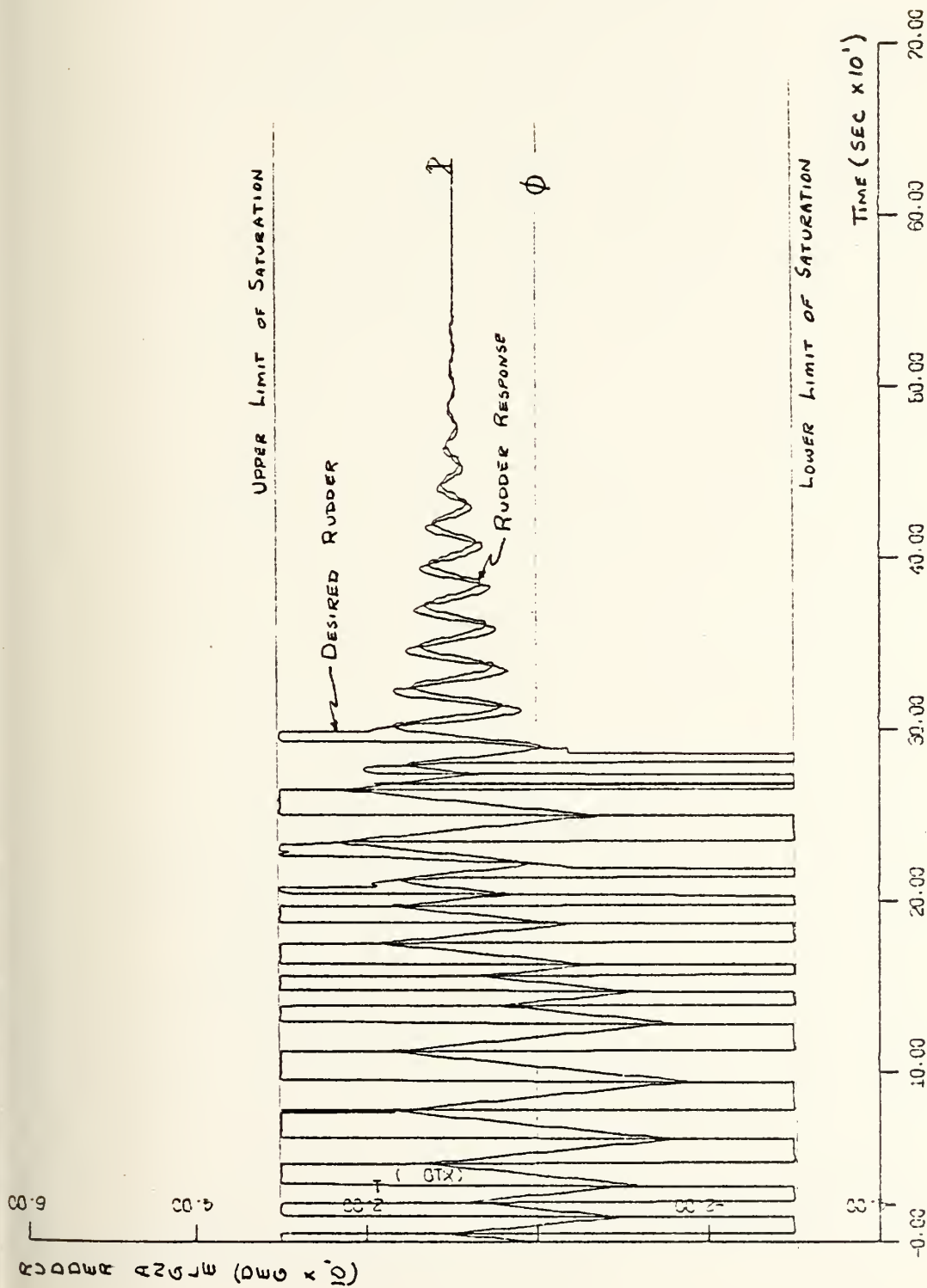


Figure III-37
Approach Phase Run #1 Rudder Response

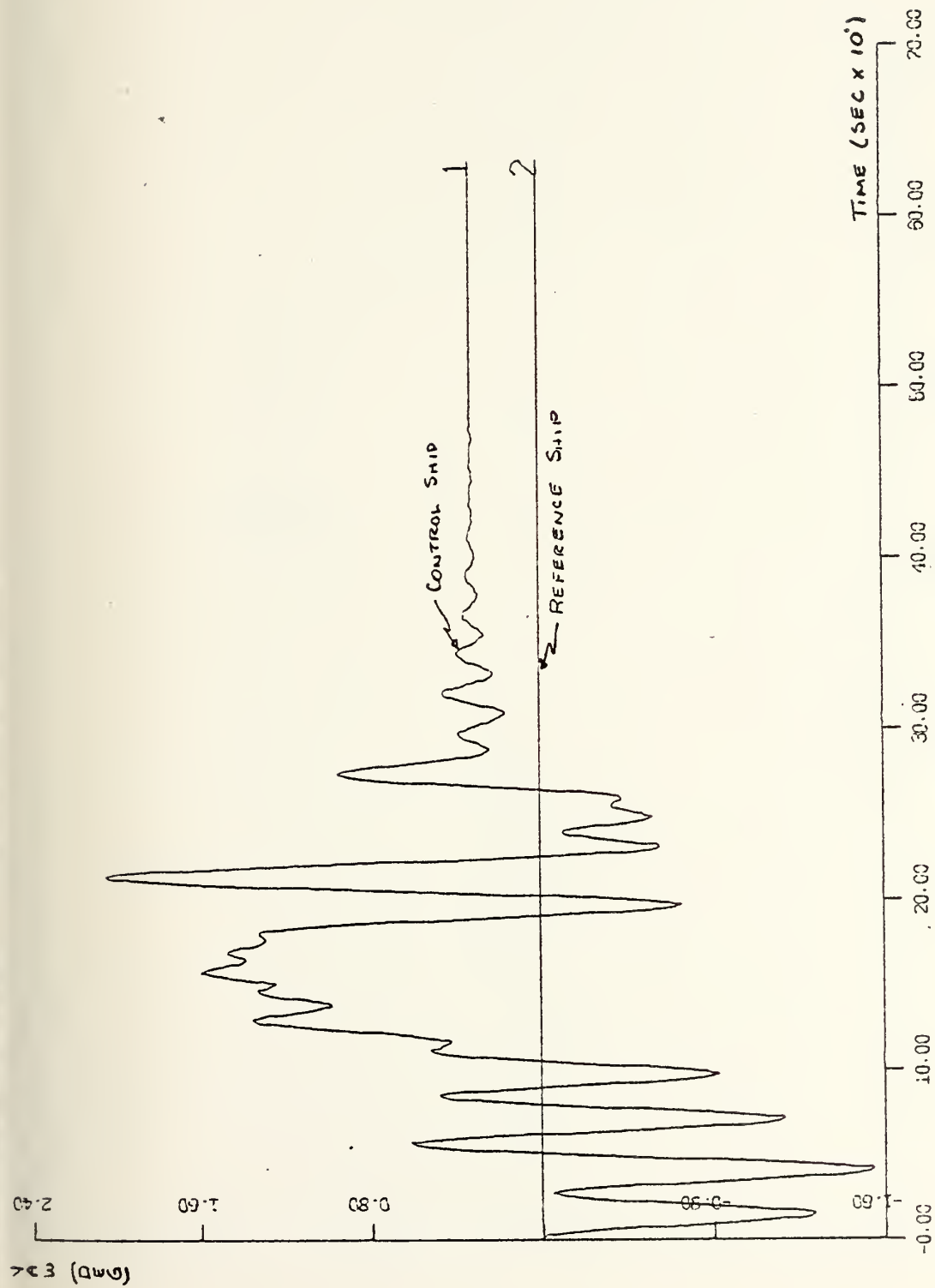


Figure III-38

Approach Phase Run #2 Yaw Response

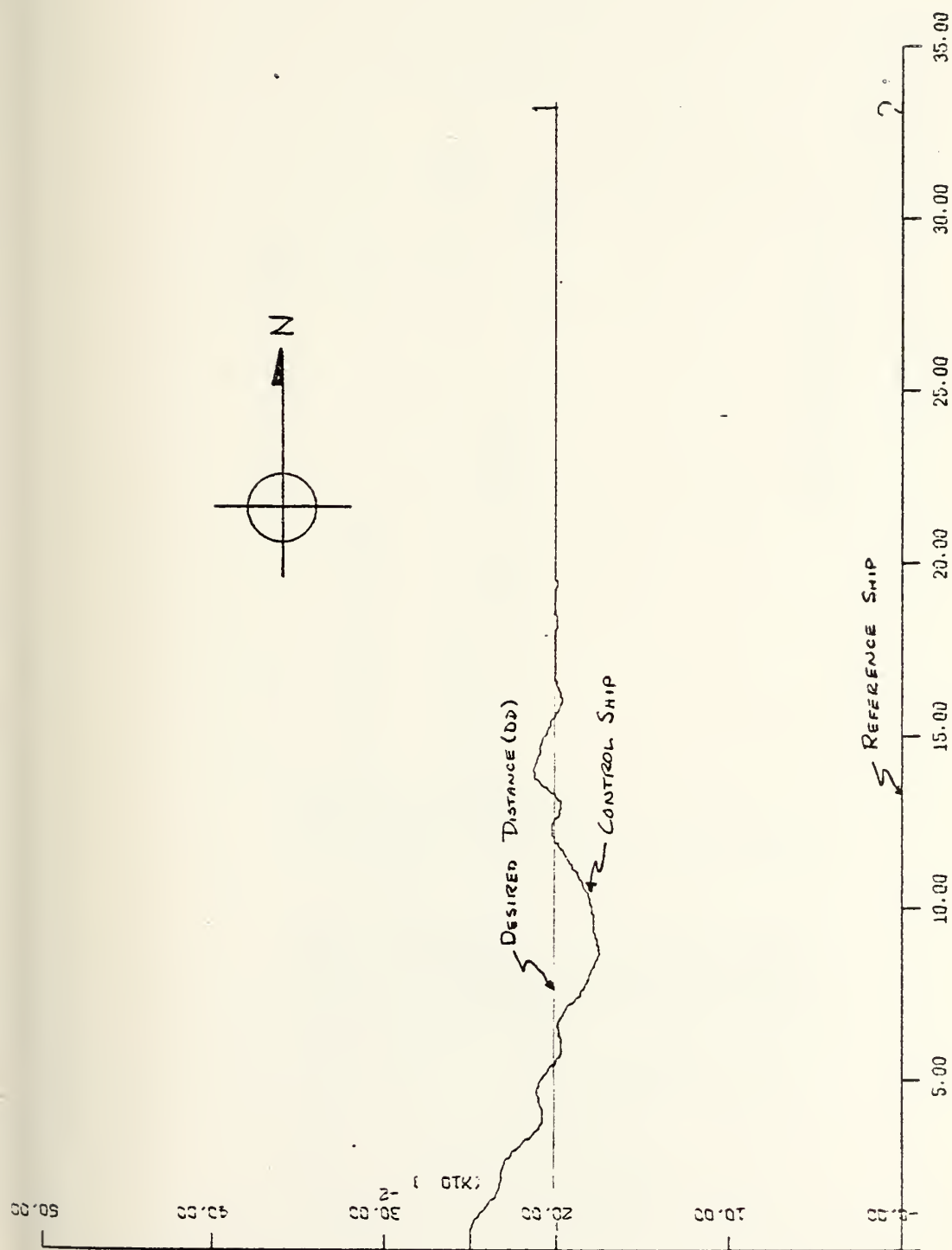


Figure III-39
Approach Phase Run #2 Geographic Plot

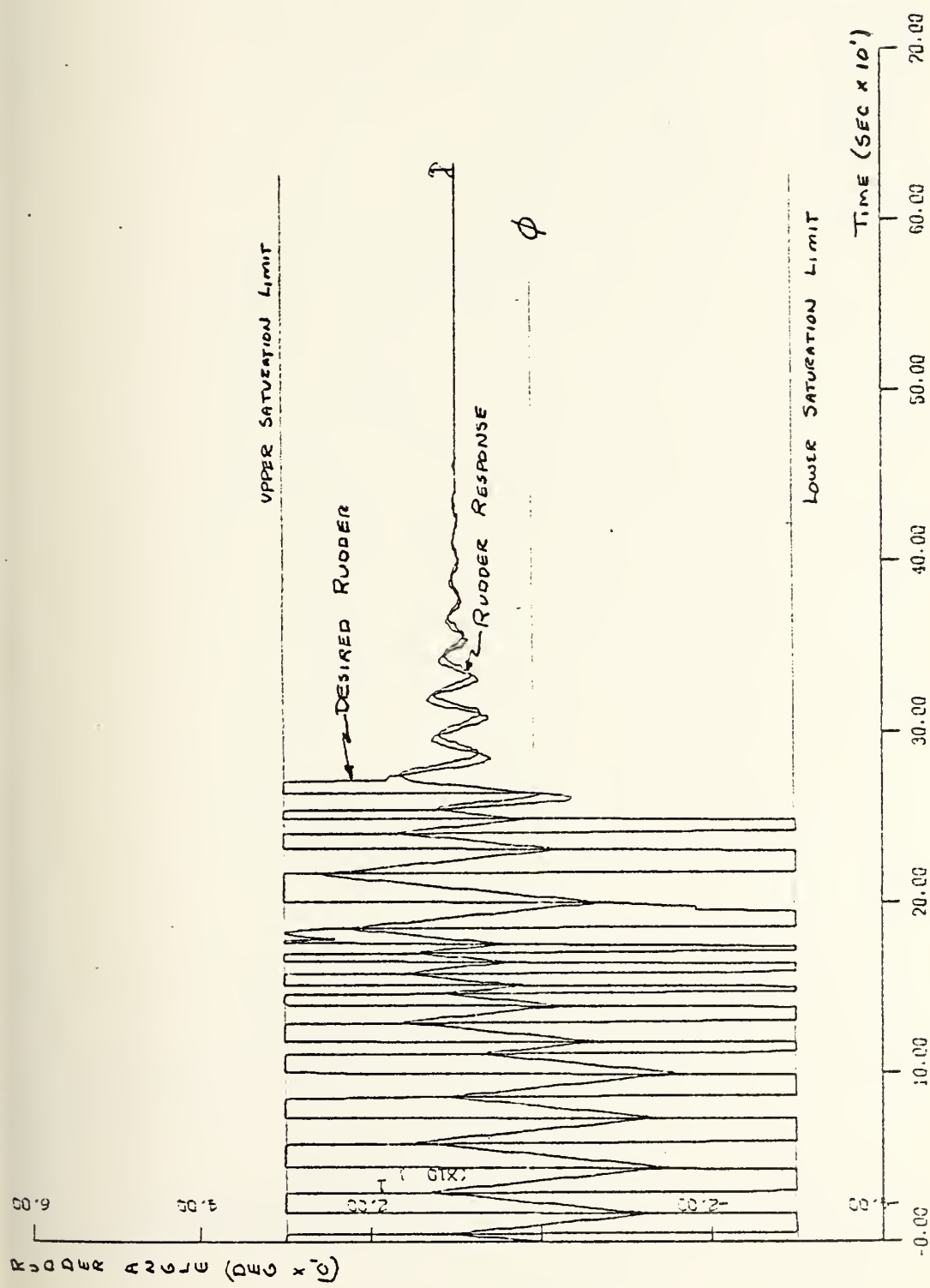


Figure III-40
Approach Phase Run #2 Rudder Response

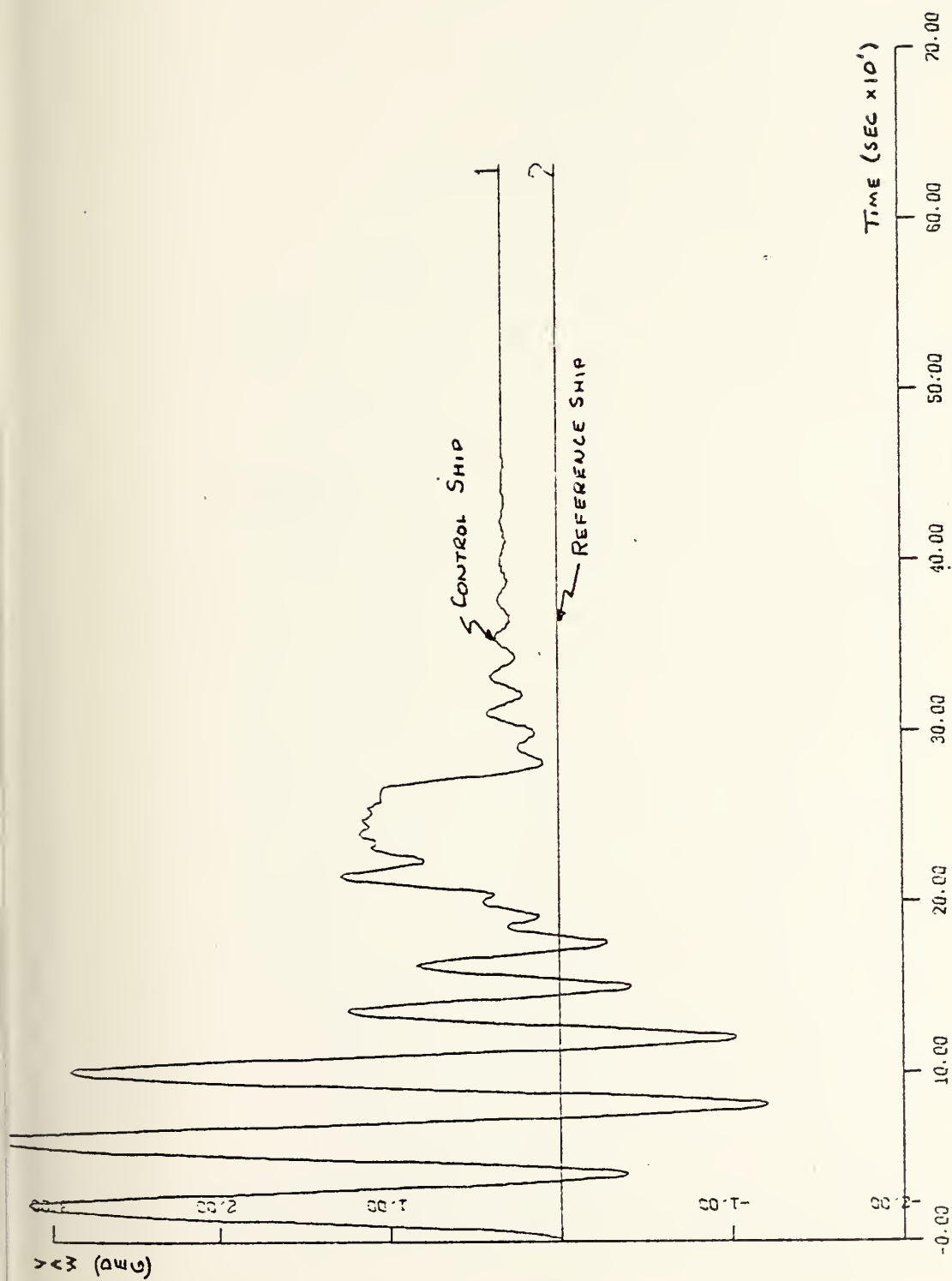


Figure III-41
Approach Phase Run #3 Yaw Response

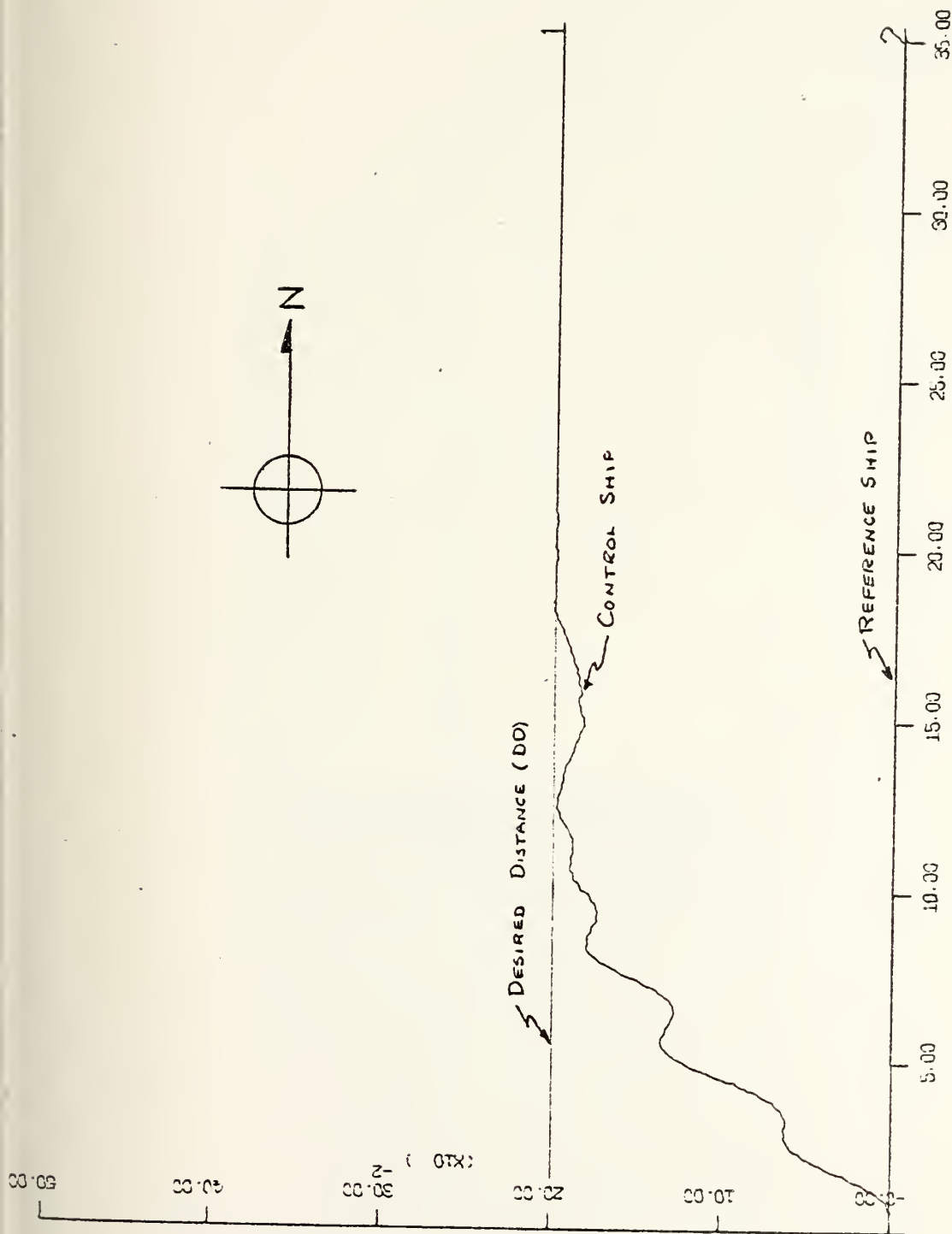


Figure III-42
Approach Phase Run #3 Geographic Plot

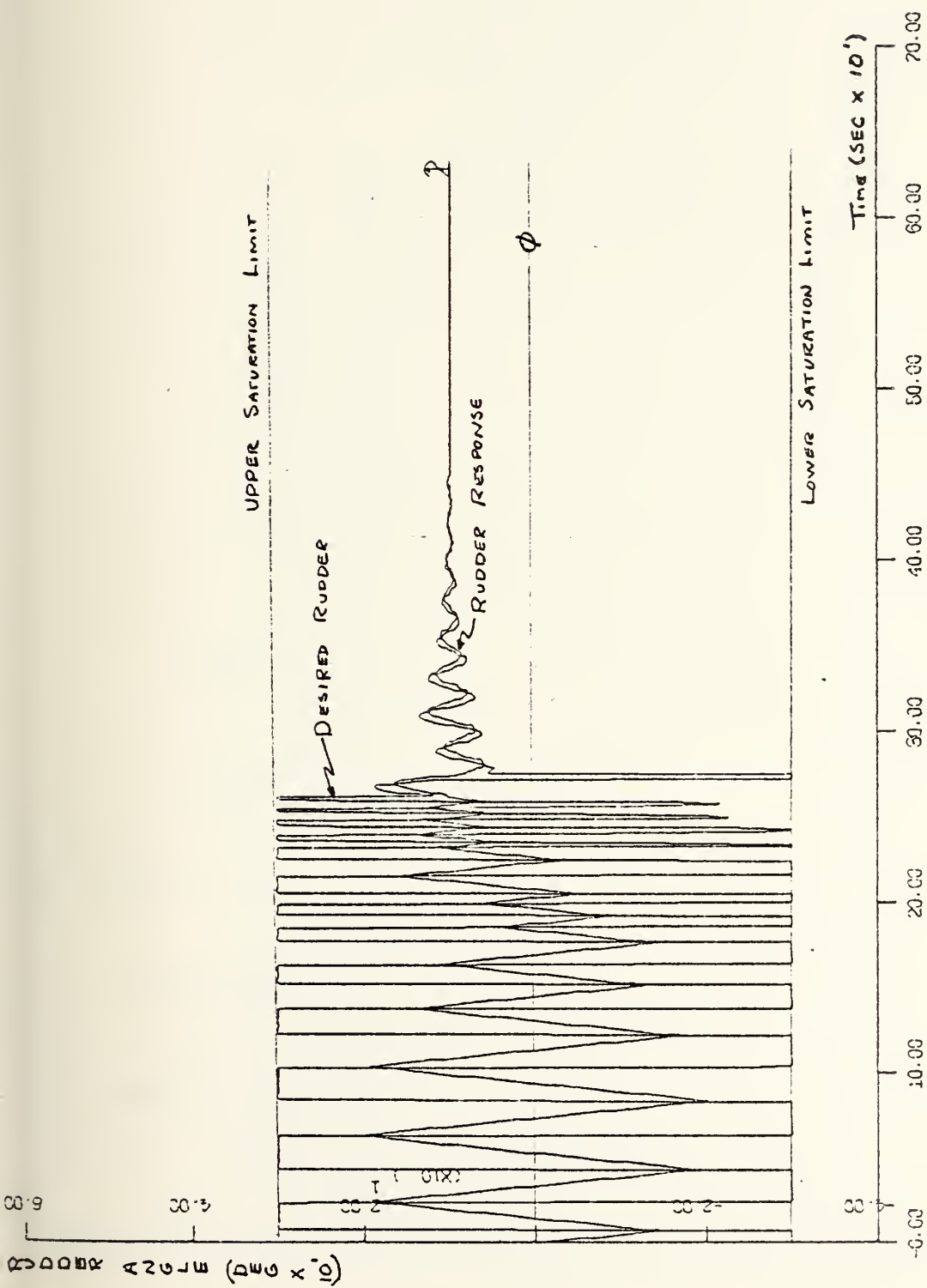


Figure III-43
Approach Phase Run #3 Rudder Response



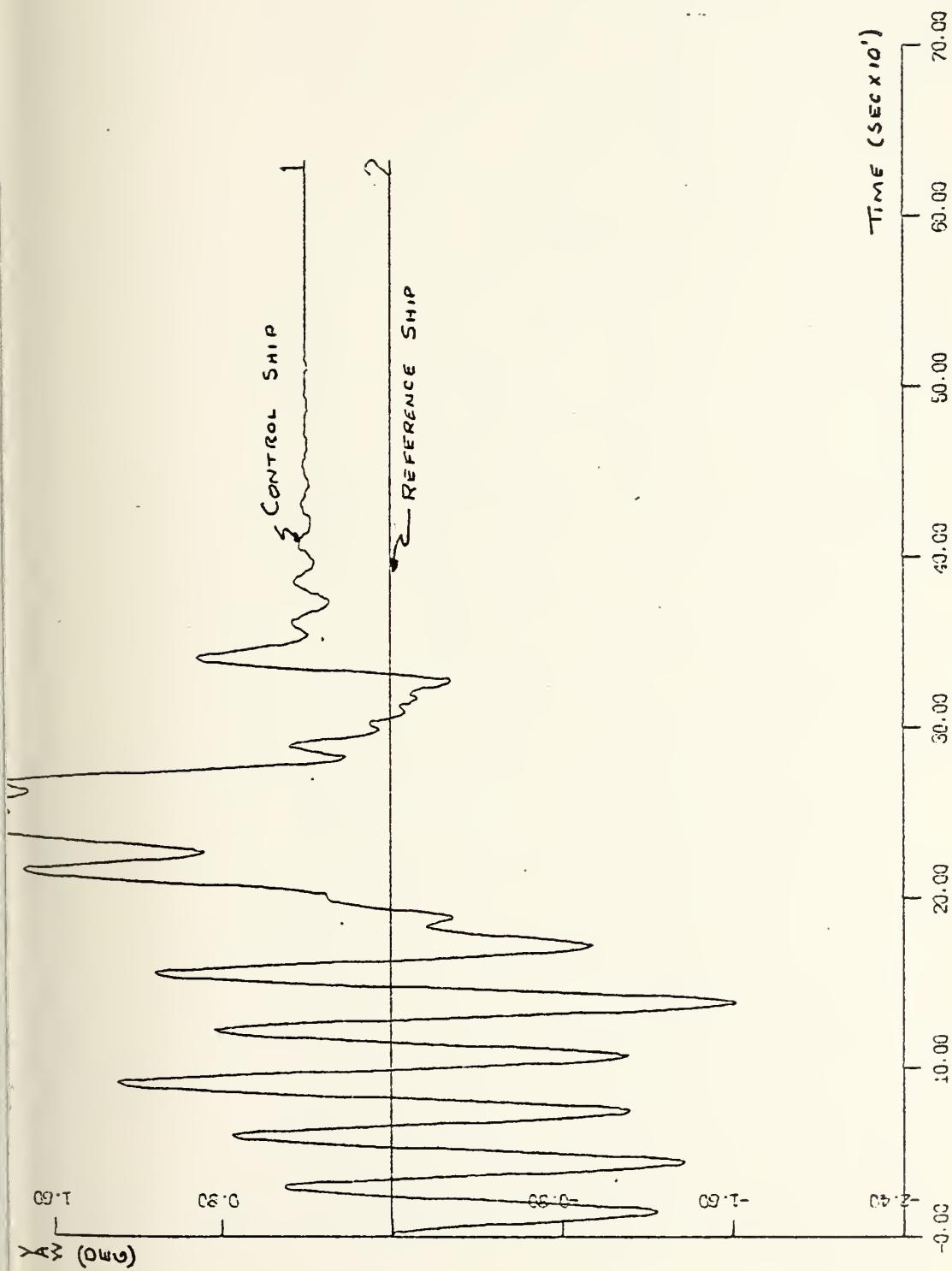


Figure III-44
Approach Phase Run #4 Yaw Response

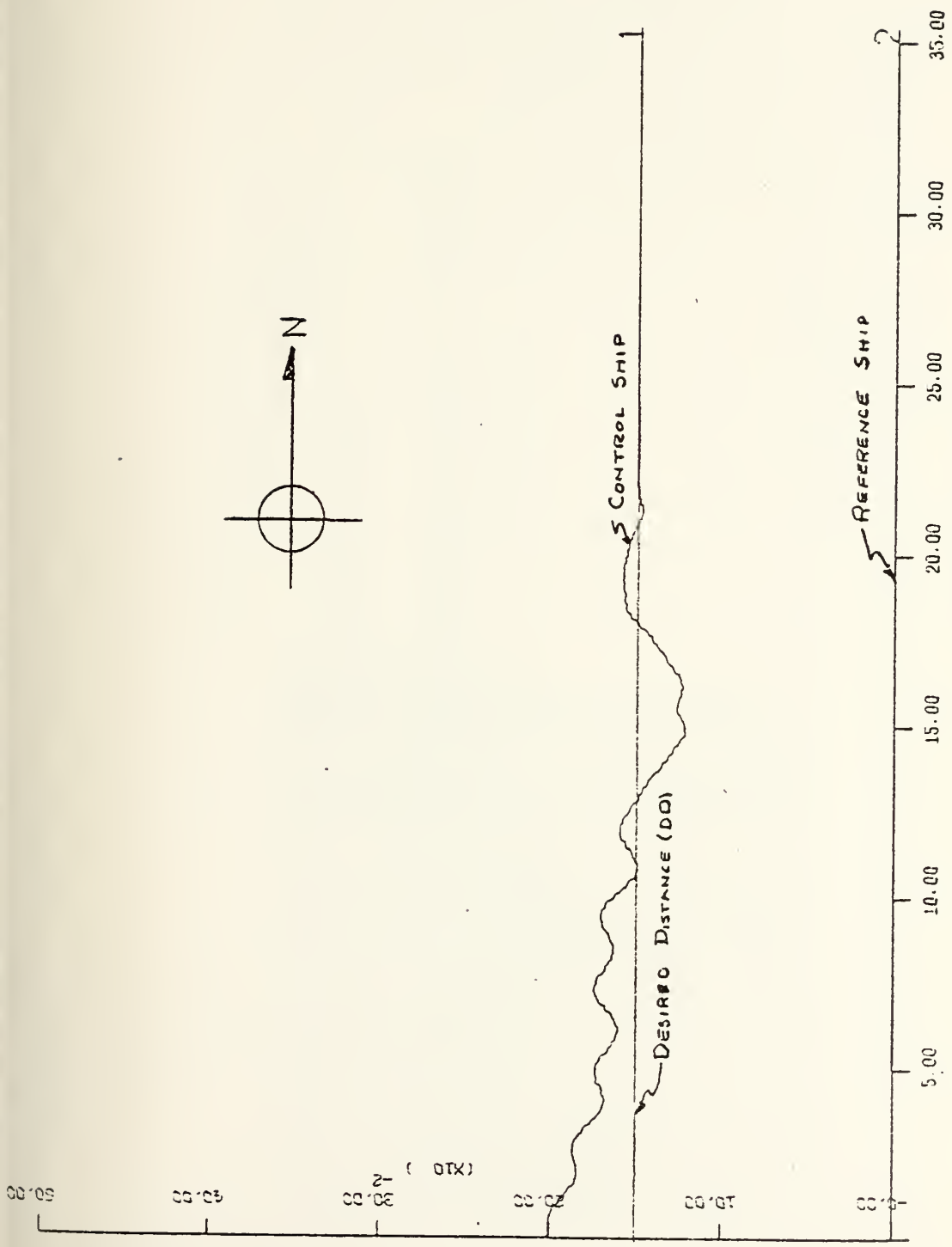


Figure III-45
Approach Phase Run #4 Geographic Plot



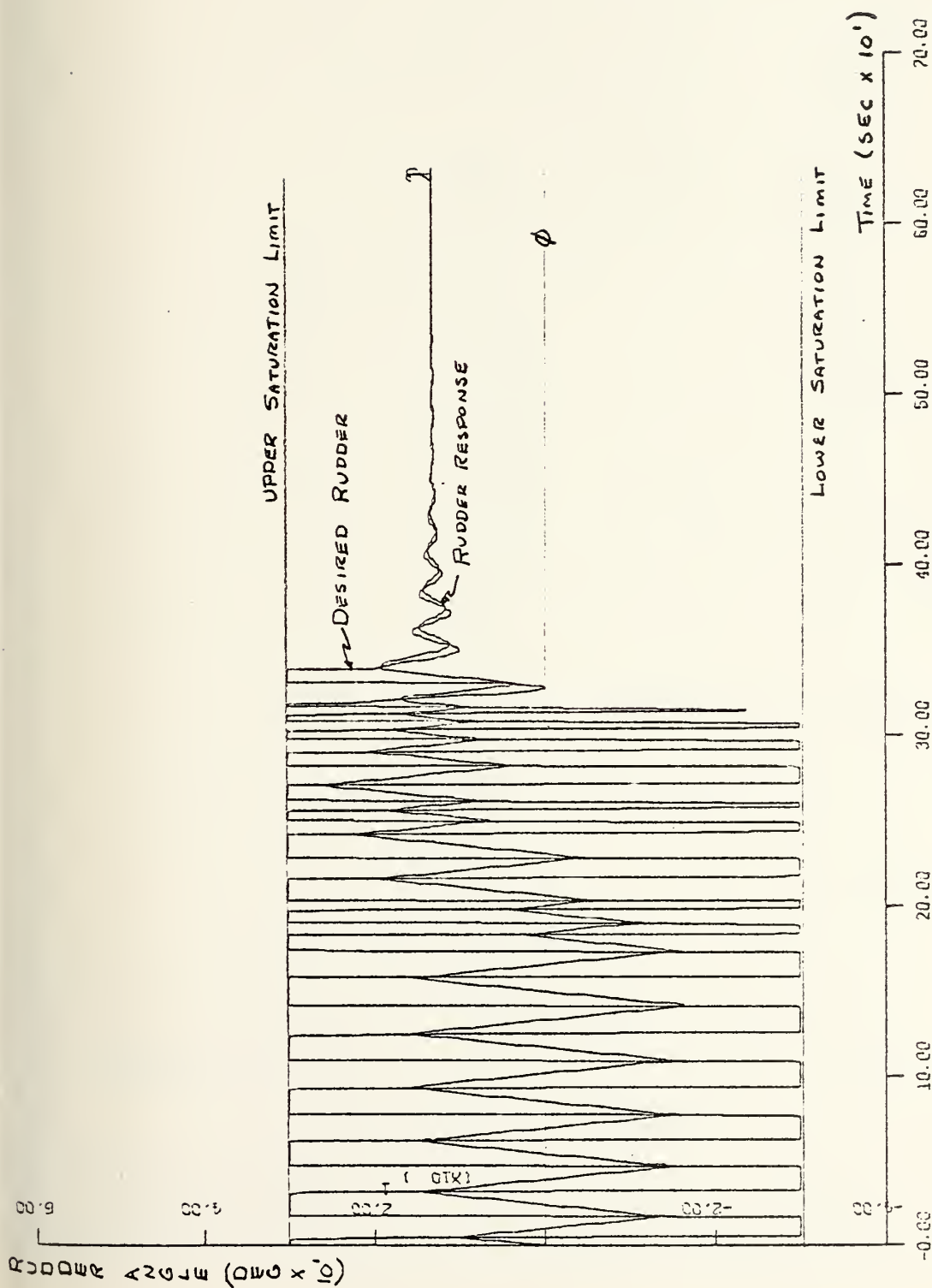


Figure III-46
Approach Phase Run #4 Rudder Response



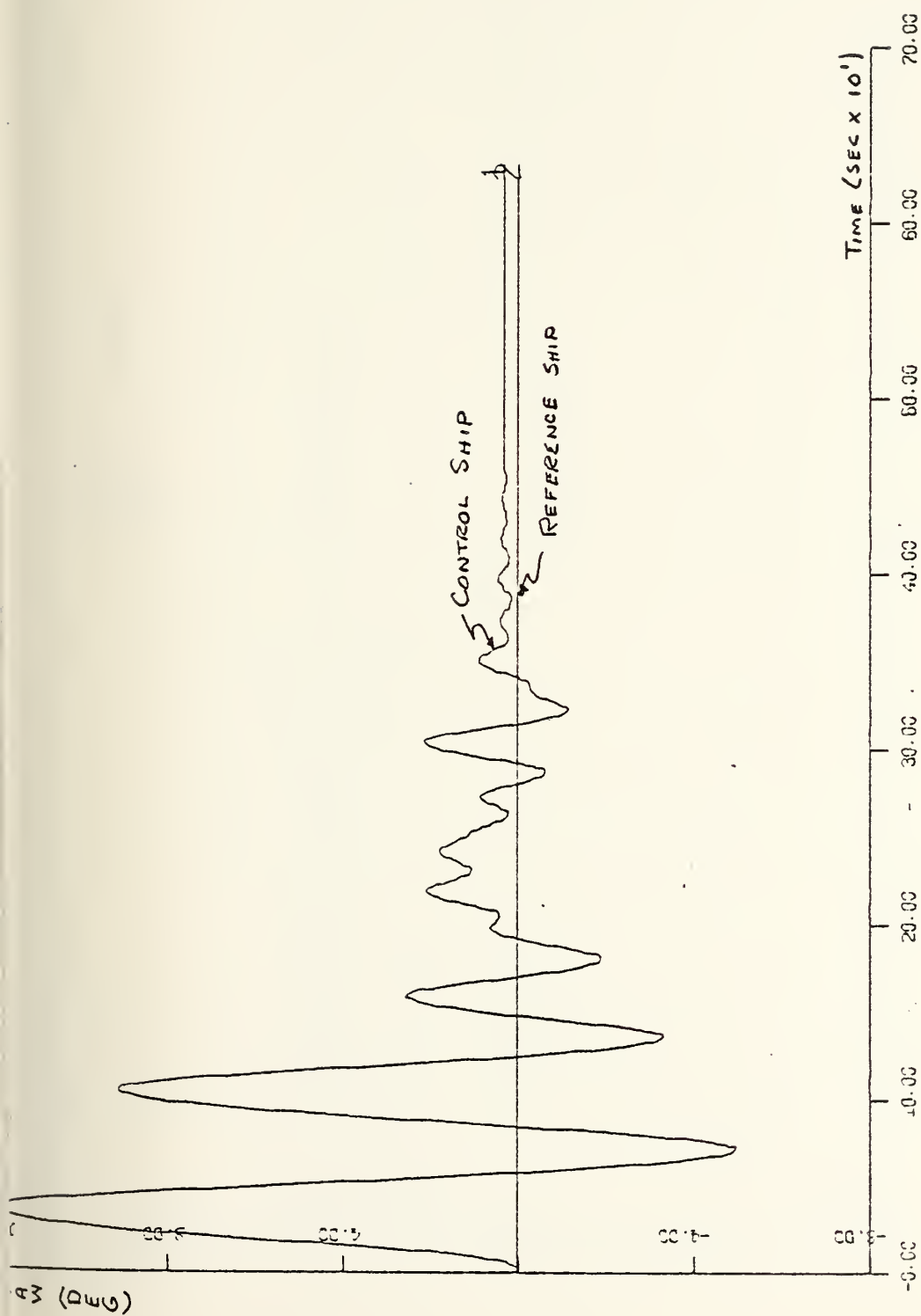


Figure III-47

Approach Phase Run #5 Yaw Response



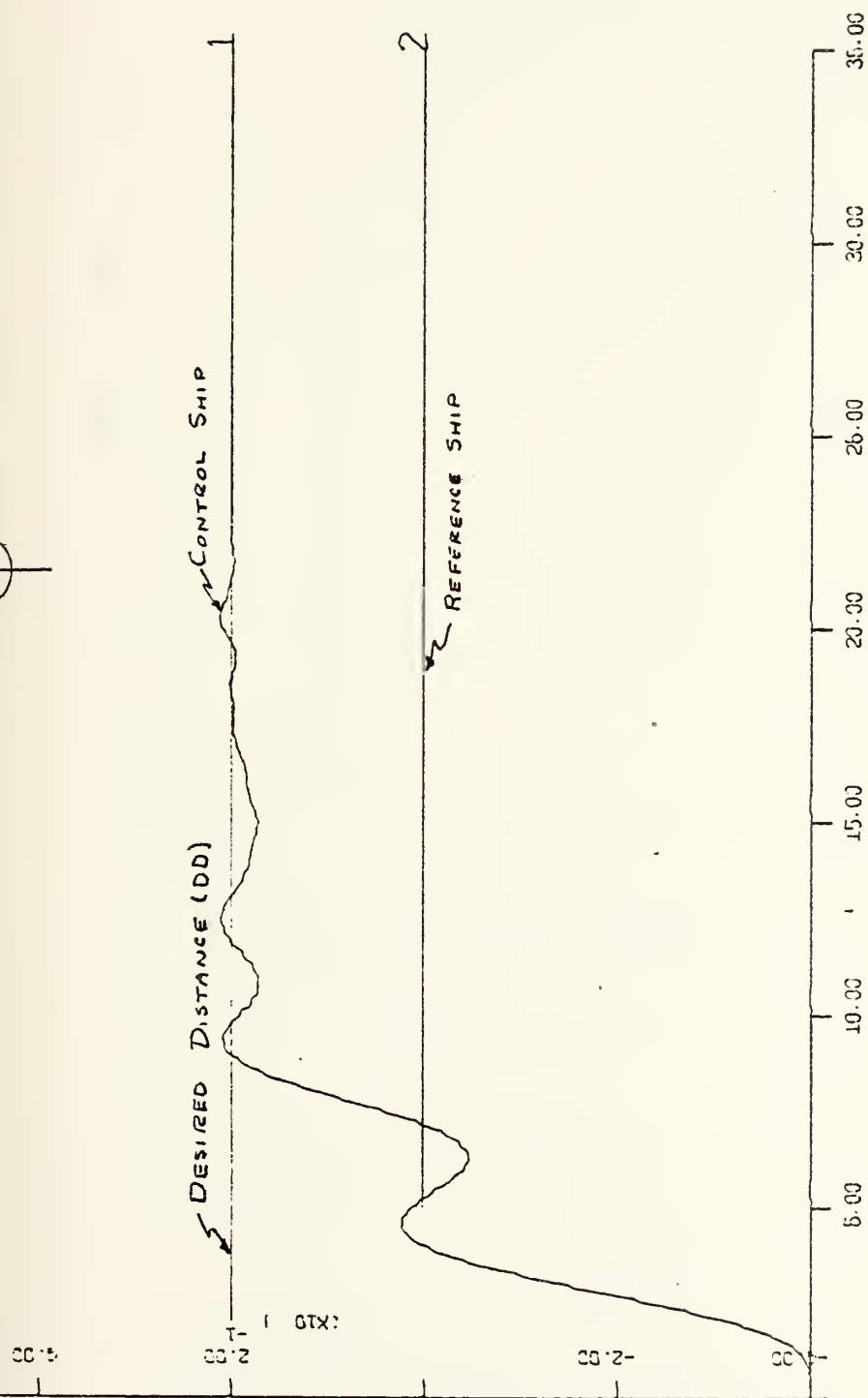
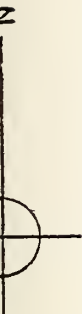


Figure III-48
Approach Phase Run #5 Geographic Plot



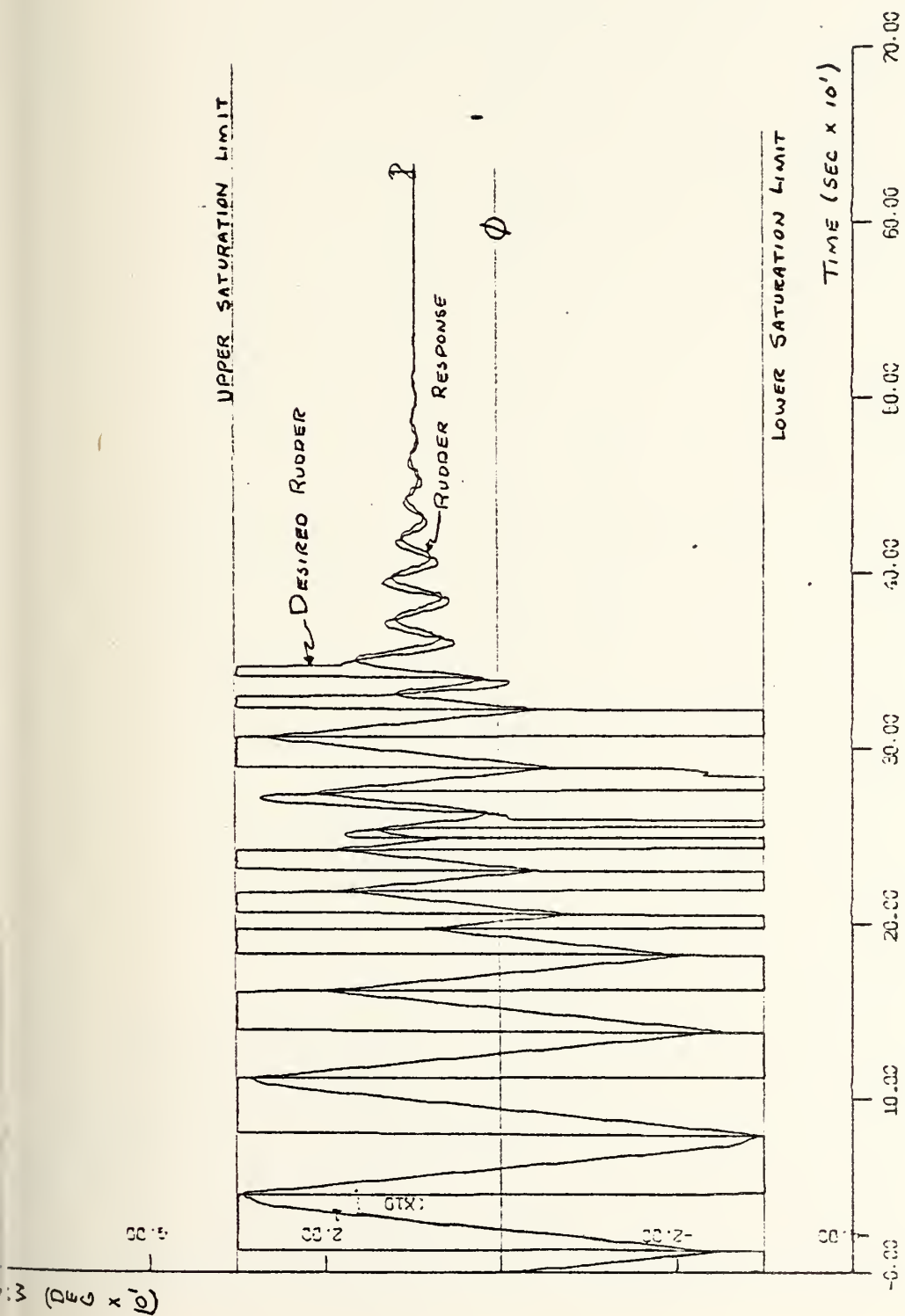


Figure III-49
Approach Phase Run #5 Rudder Response



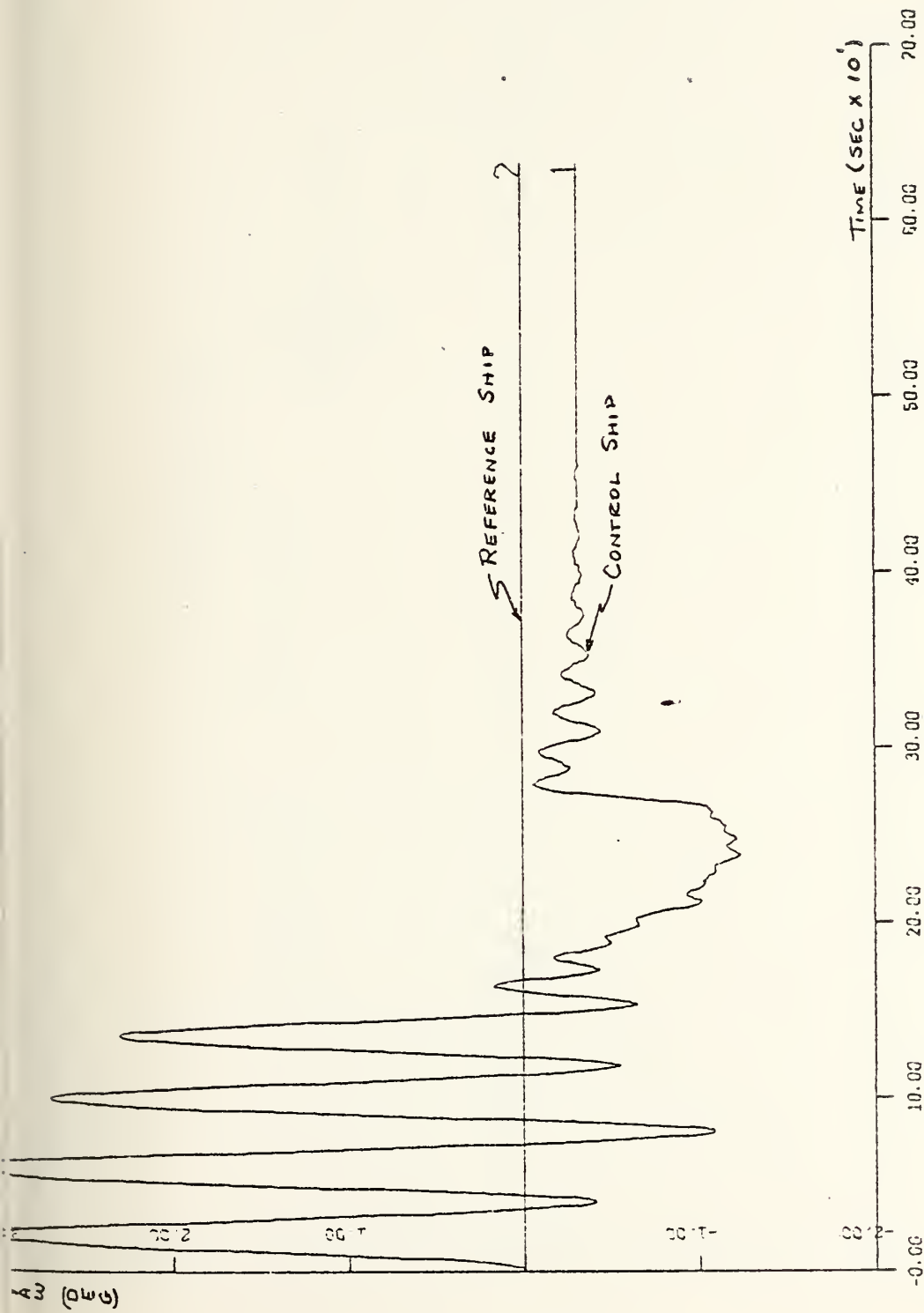


Figure III-50
Approach Phase Run #6 Yaw Response



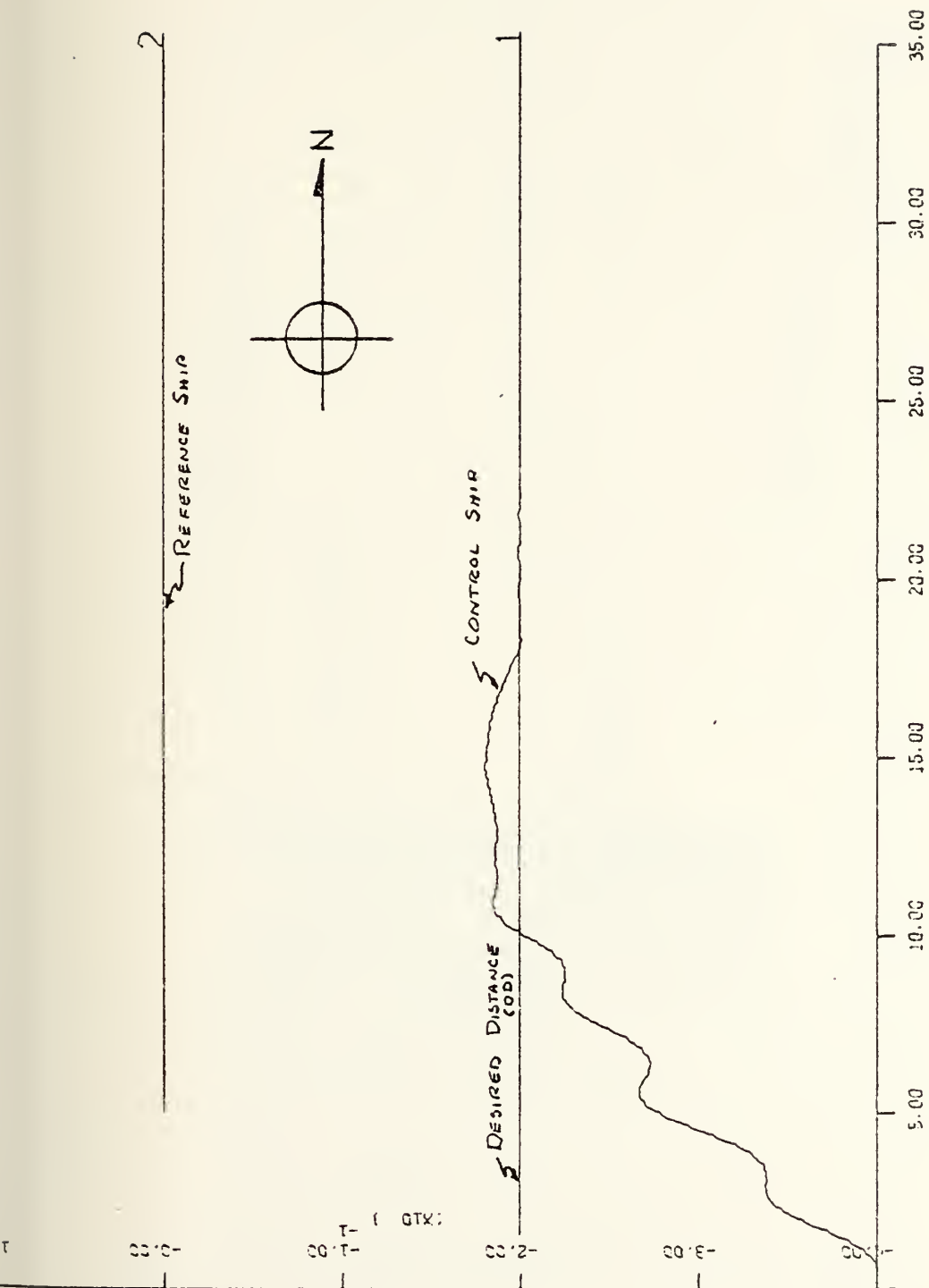


Figure III-51
Approach Phase Run #6 Geographic Plot



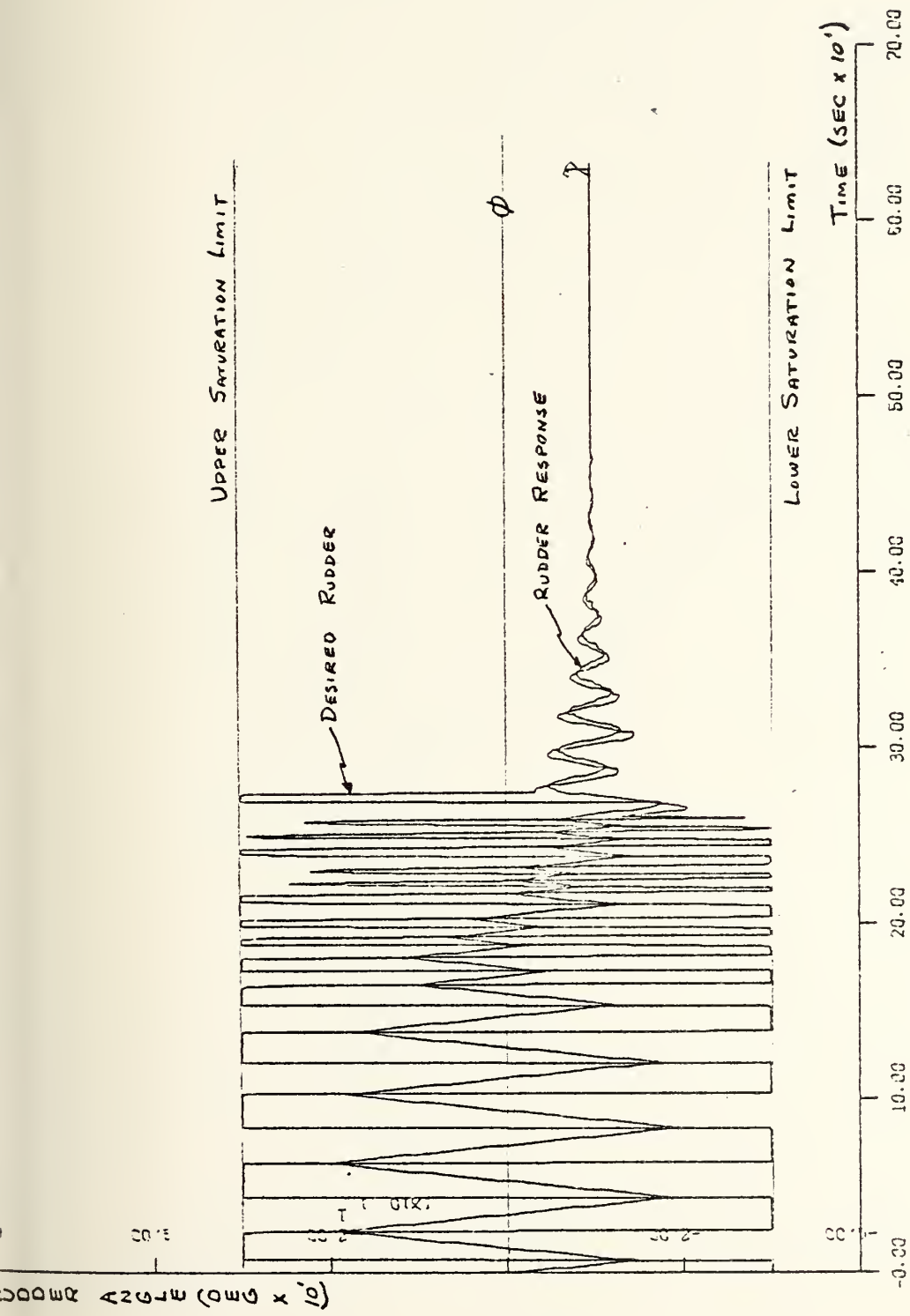


Figure III-52
Approach Phase Run #6 Rudder Response



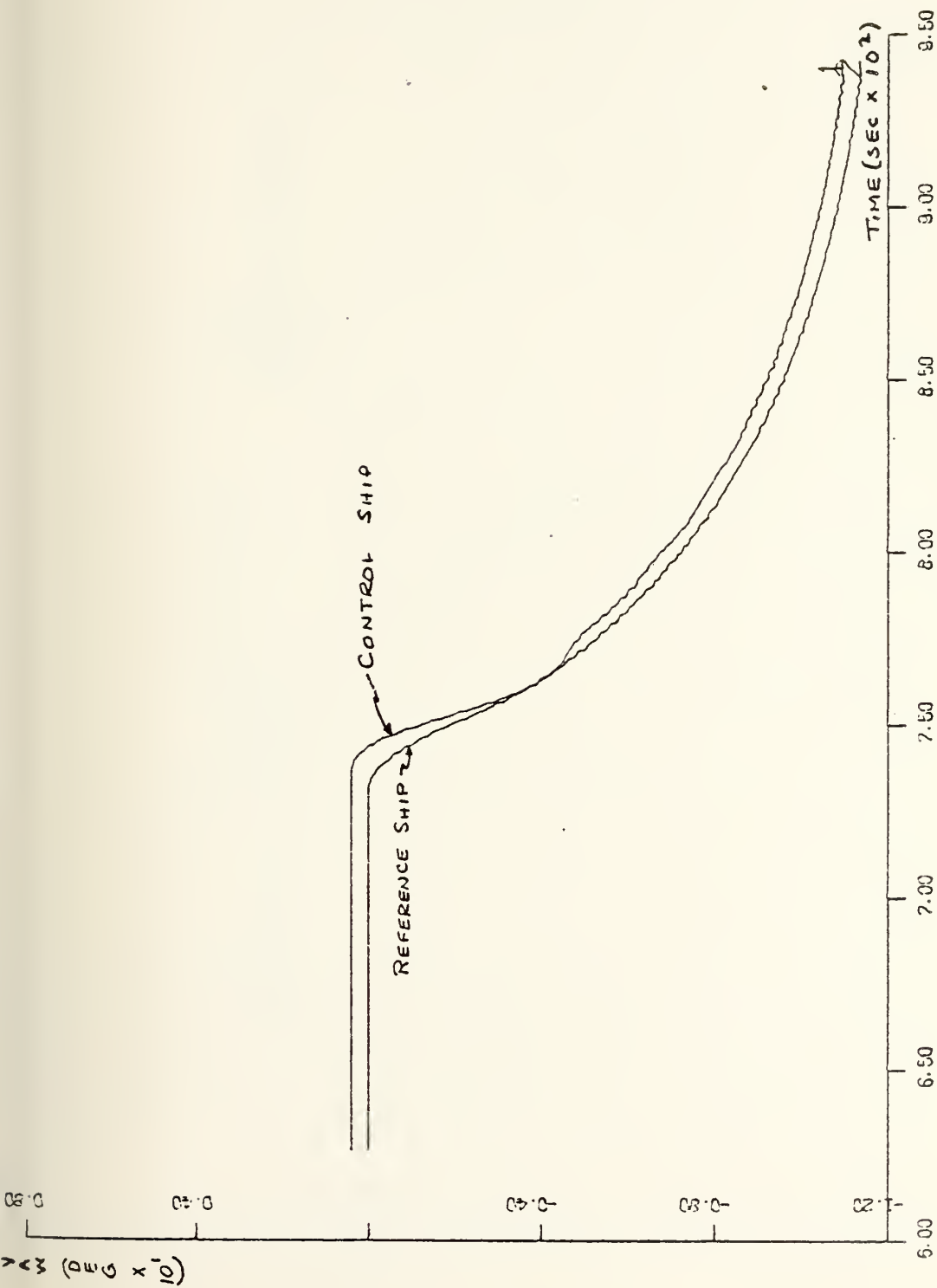


Figure III-53
Turn Phase Run #4 Yaw Response



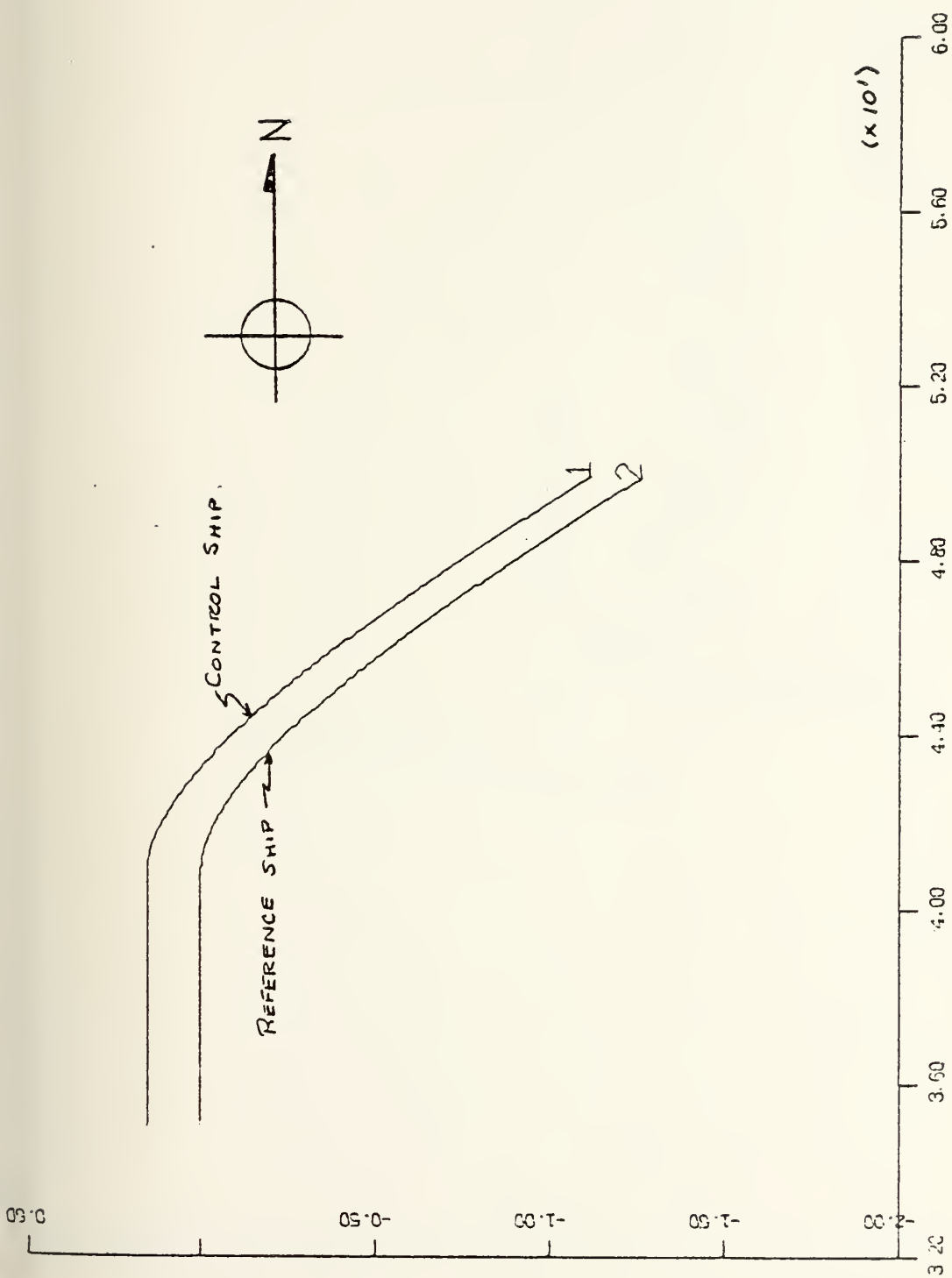


Figure III-54
Turn Phase Run #4 Geographic Plot



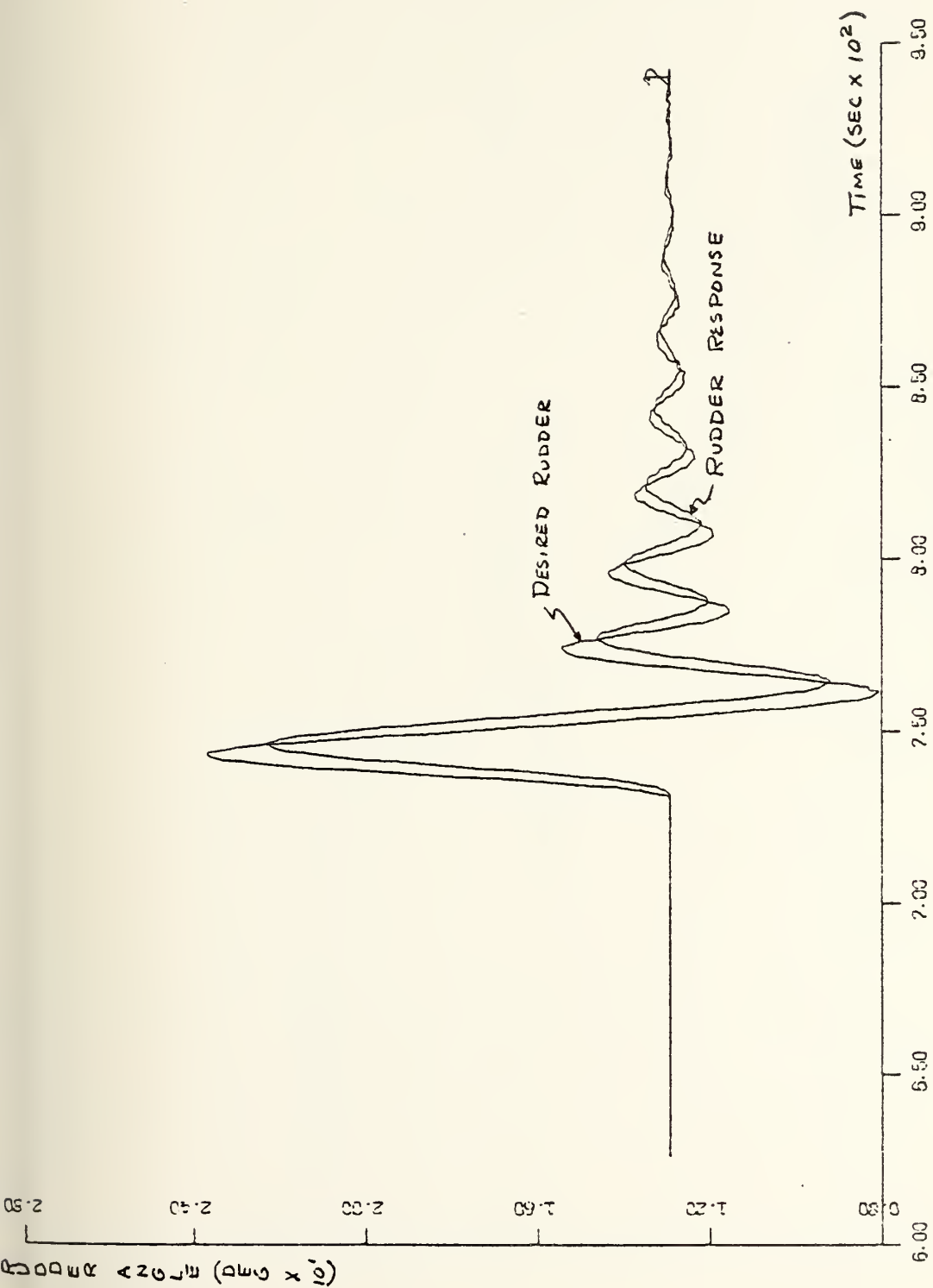


Figure III-55
Turn Phase Run #4 Rudder Response



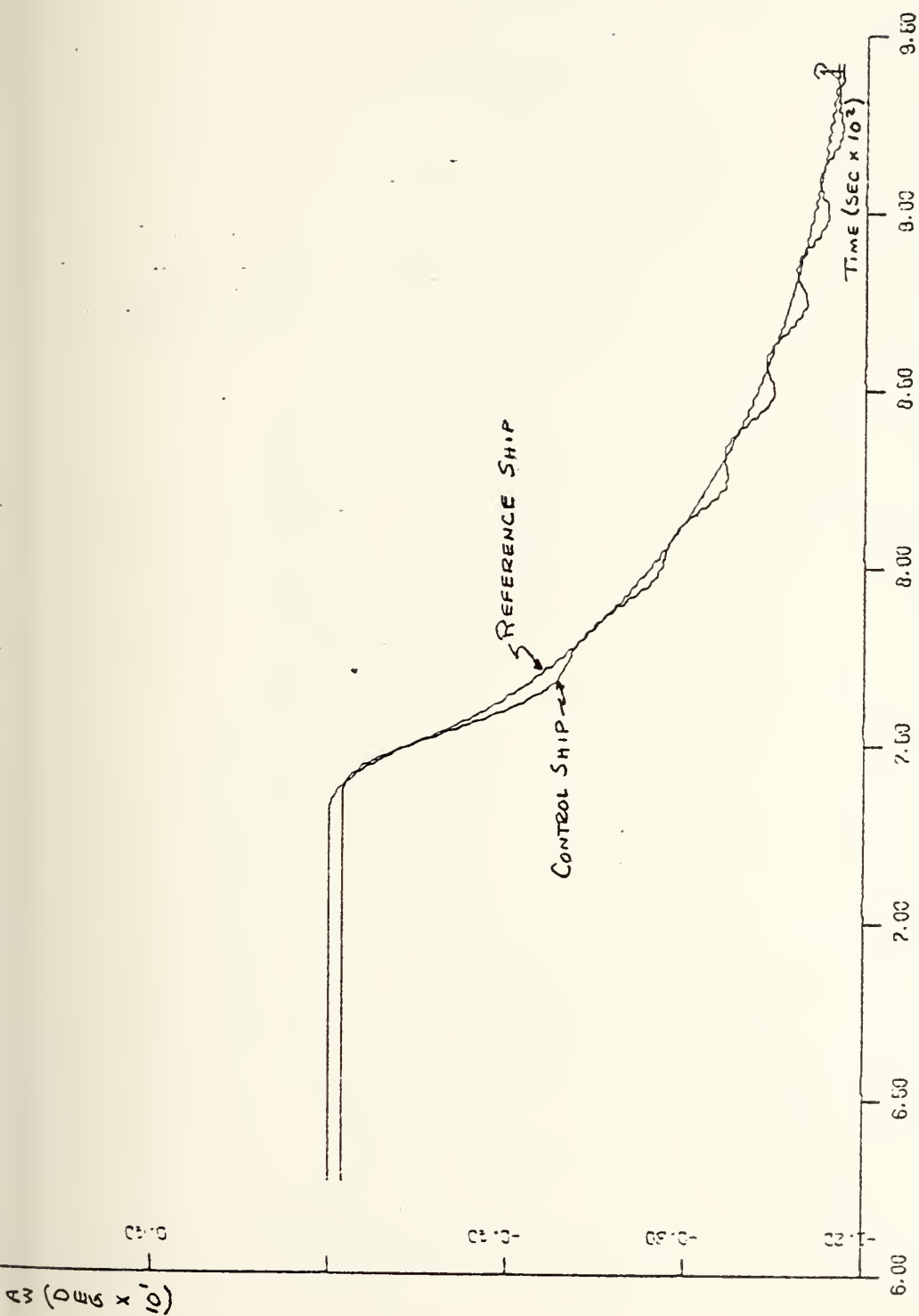


Figure III-56
Turn Phase Run #6 Yaw Response



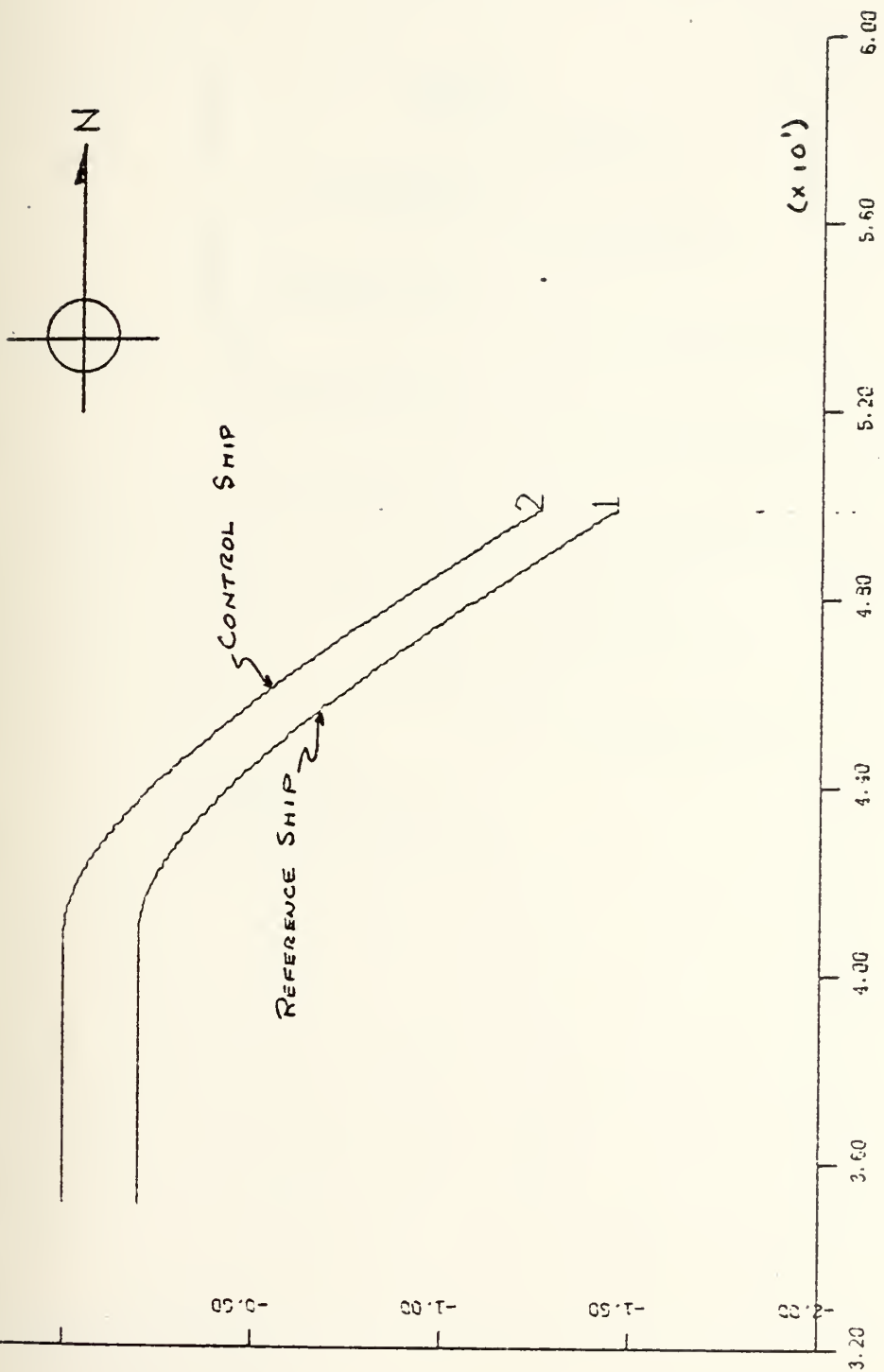


Figure III-57
Turn Phase Run #6 Geographic Plot



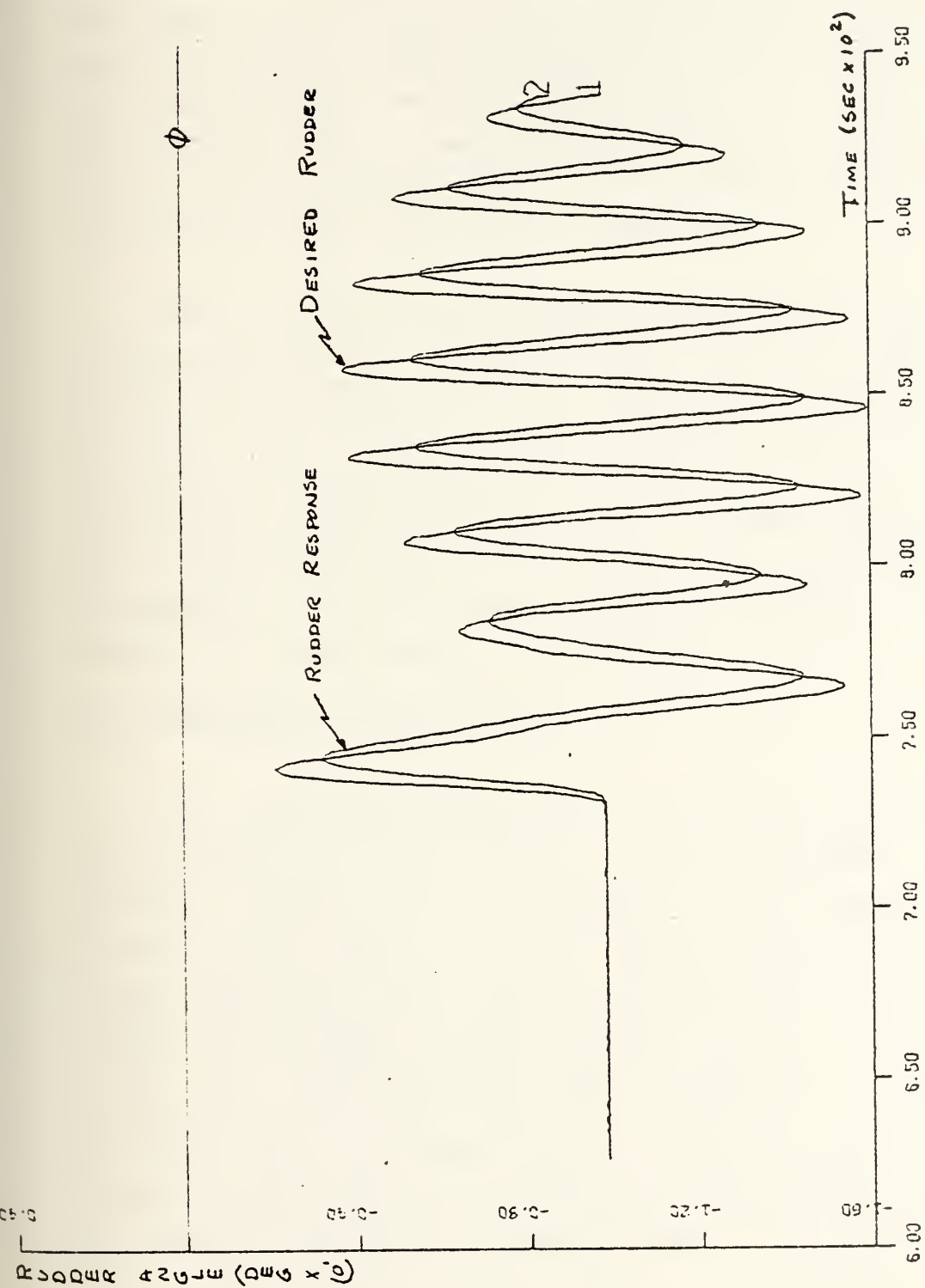


Figure III-58
Turn Phase Run #6 Rudder Response

experienced which gives 1.6 ship lengths bow to stern clearance).

The purpose of run 6 is to provide simulation for an approach from the opposite side again disproving any concern for ambiguity in the trigonometric measurement scheme utilized. In all runs it must be emphasized that DD is the positive absolute distance desired and that IS provides the code flag for the desired side of approach. The system will work with DD set to some negative quantity; but the side of approach will reverse itself and the position placement will be correct, but on the side not desired.

Run 4 takes the desired distance in to 0.15 ship lengths (80.0 feet). This distance is usually the minimum desired by a prudent seaman. Again, even with this minimum distance, the control system performs up to desired standards. The importance of this run cannot be overlooked. Performance of the system at this extremum indicates that the gains utilized are correct for all expected conditions encountered in calm seas. Figures III-59 thru III-64 portray the remaining plots obtained in run 4.

h. Performance in Sea State

The calm sea performance of the heading control system is only part of the system testing required. Of even greater concern is the adequacy of the control when sea state is introduced. Section D.2. of chapter II models the three components of waves with two sinusoids and a small random impulse wave. These forces were introduced into the total RAS simulation as shown in computer program #7. In this program the wave length (WL) is set to one ship length and the wave direction (WD) is -015 degrees true. This scenario allows for a port turn into the prevailing sea as is common practice in experienced RAS evolutions. By



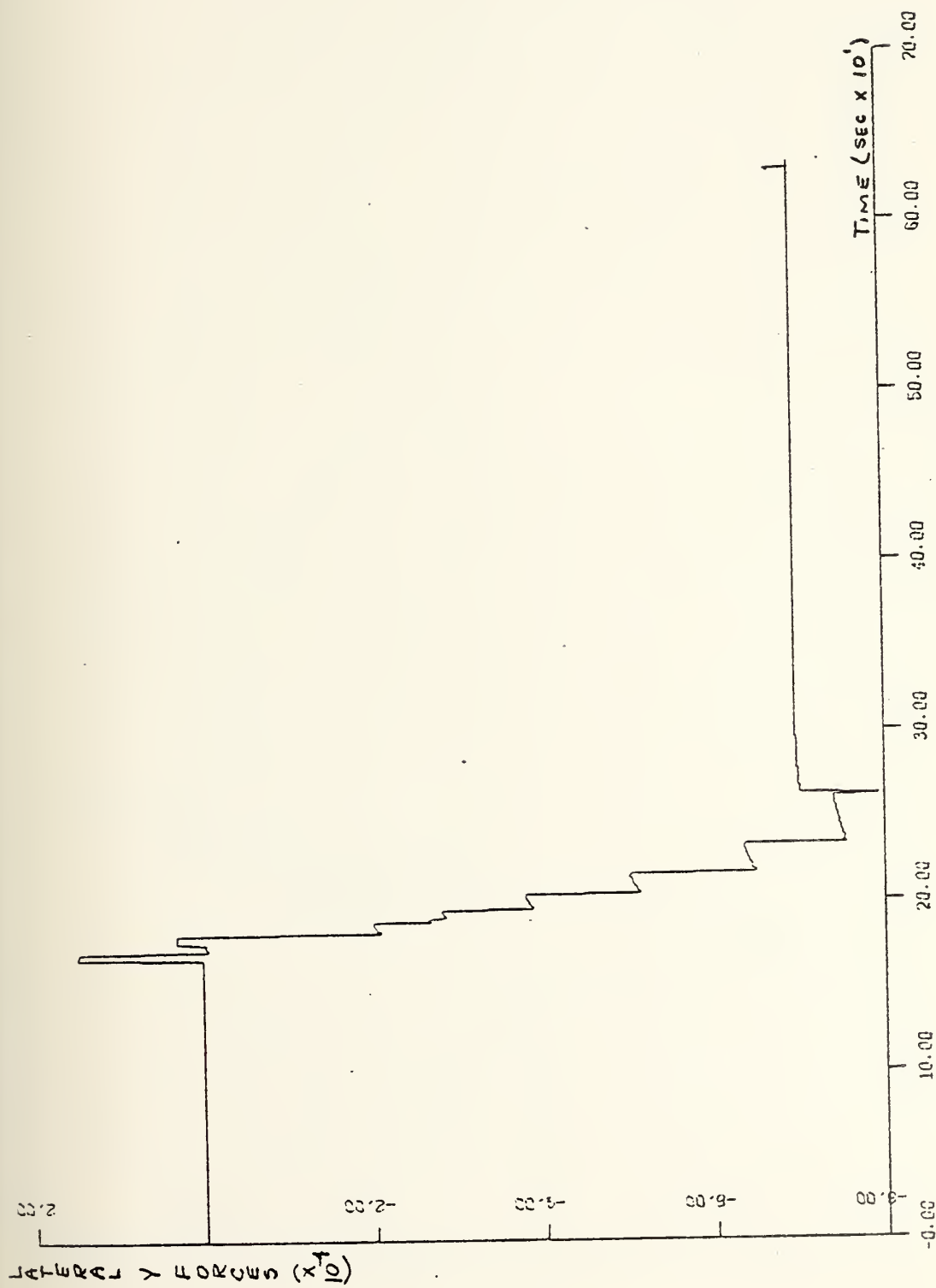


Figure III-59
Approach Phase Run #4 Lateral Y Forces



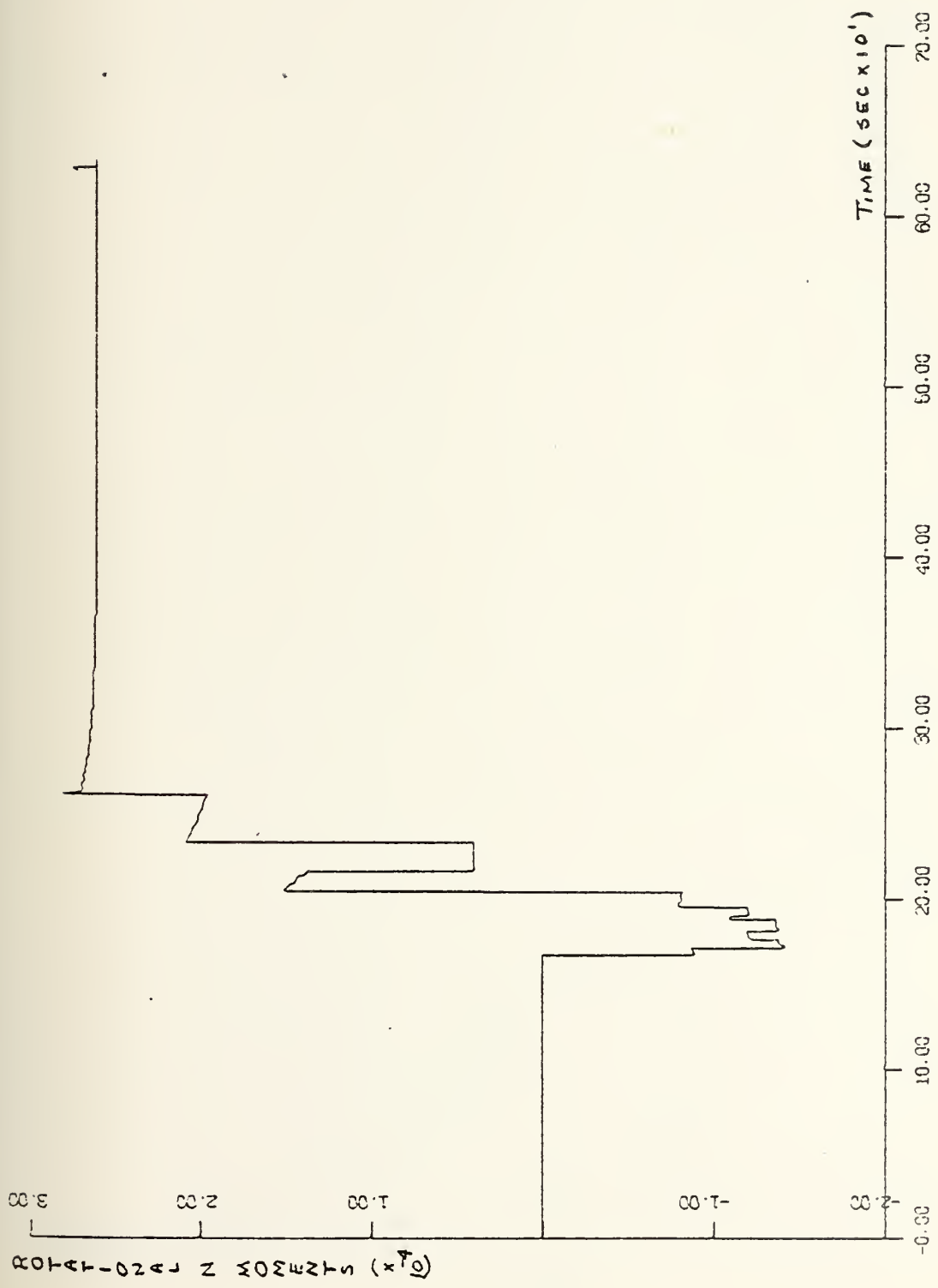


Figure III-60
Approach Phase Run #4 Rotational N Moments



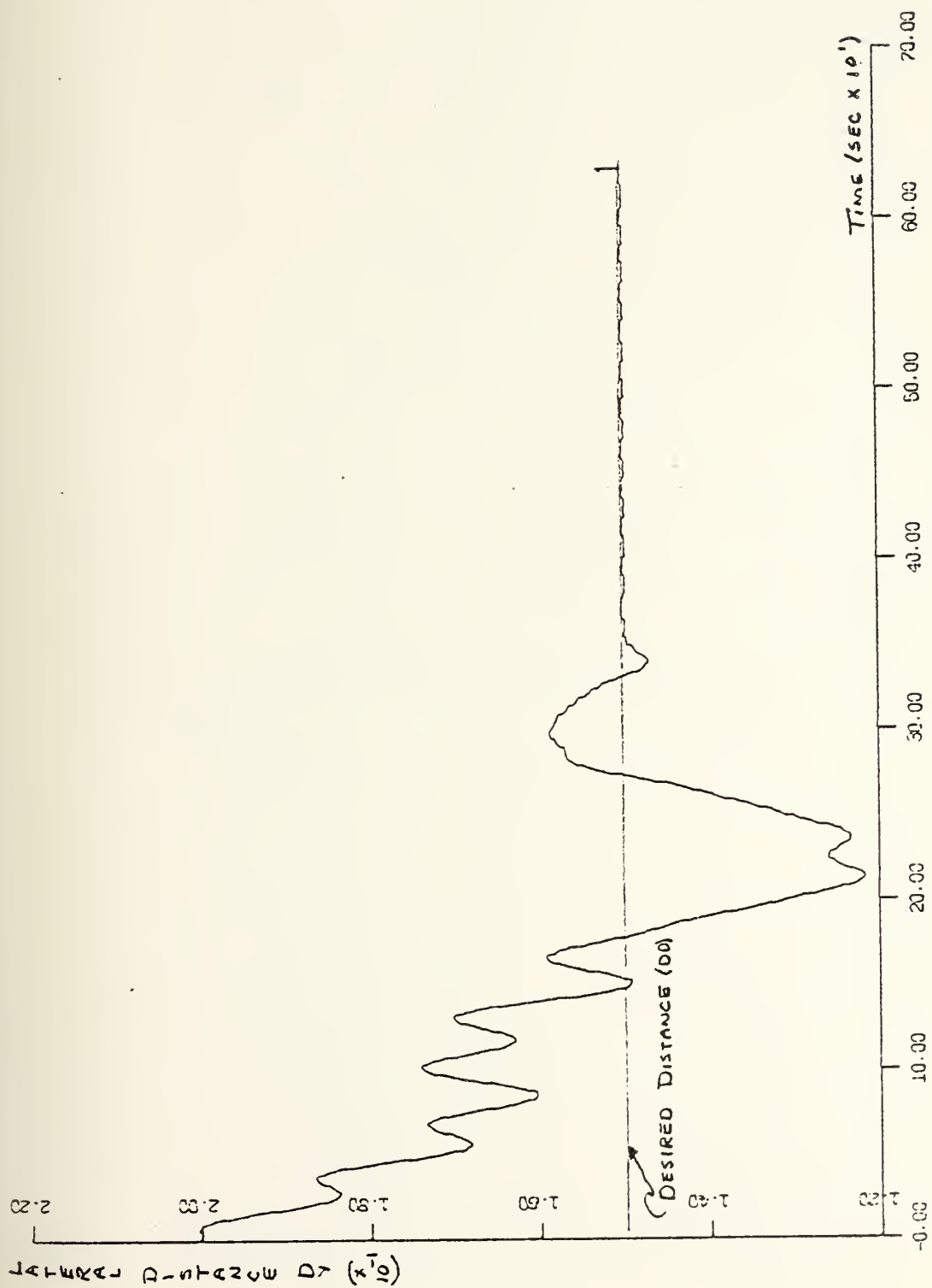


Figure III-61
Approach Phase Run #4 Lateral Distance DY



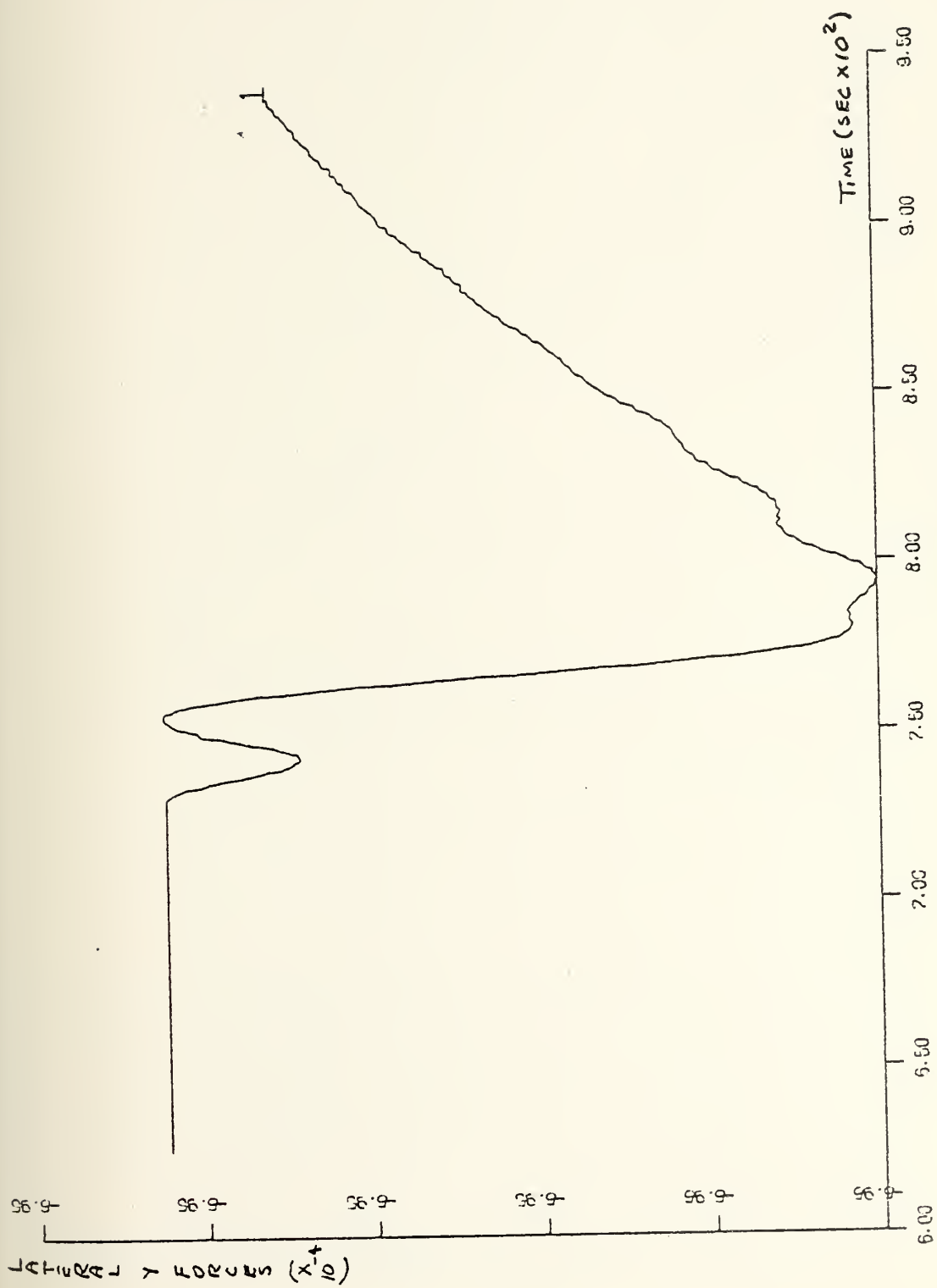


Figure III-62
Turn Phase Run #4 Lateral Y Forces



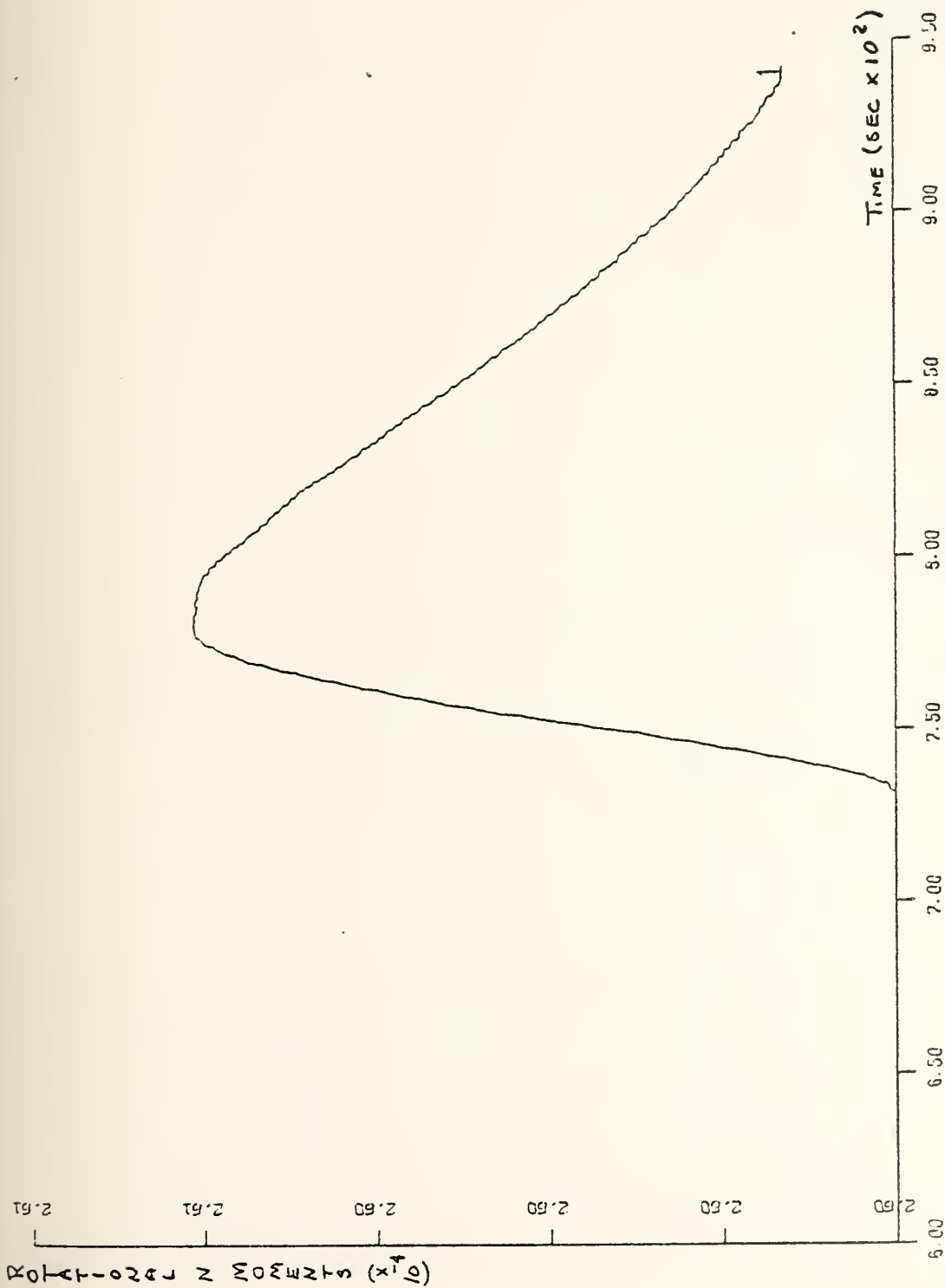


Figure III-63
Turn Phase Run #4 Rotational N Moments



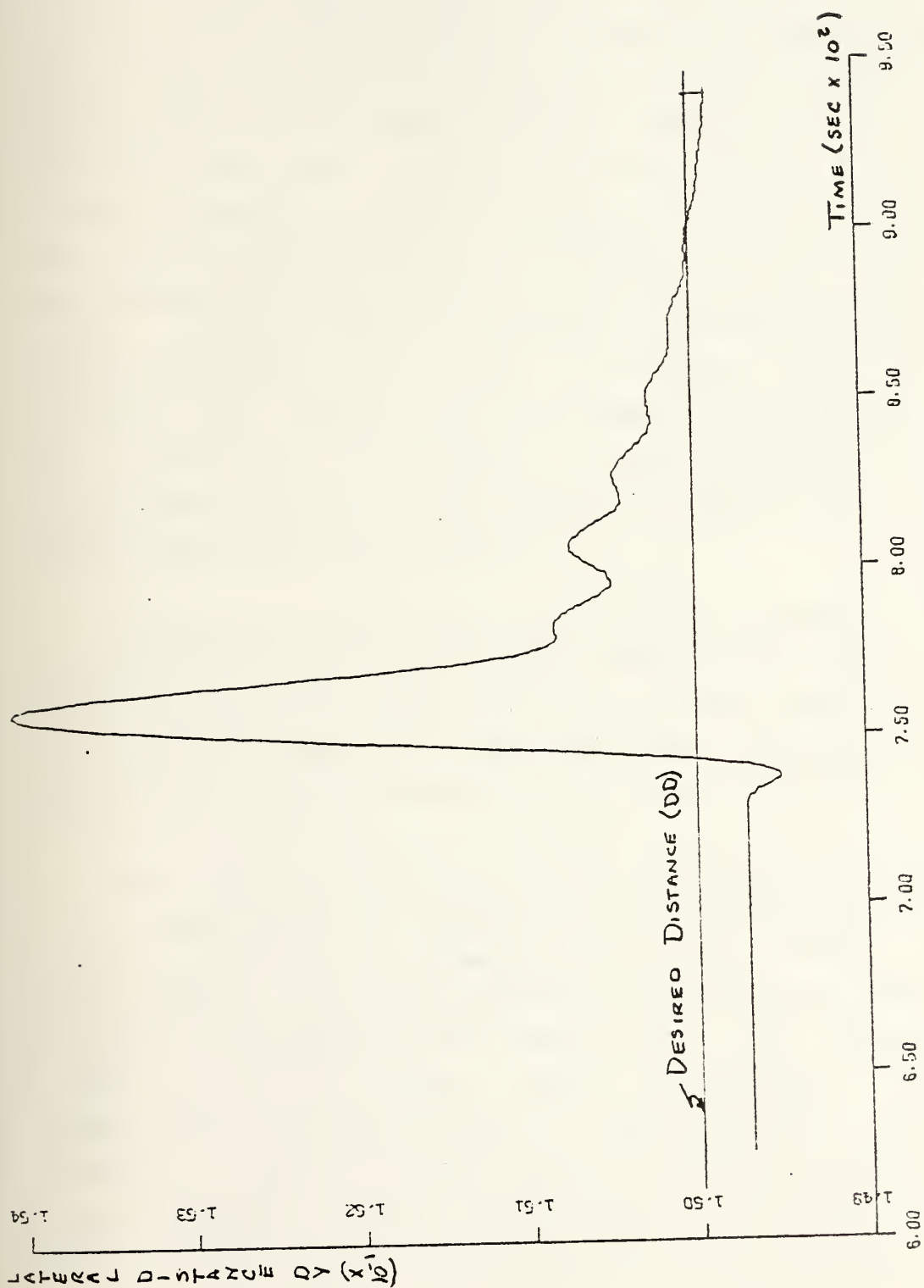


Figure III-64
Turn Phase Run #4 Lateral Distance DY



minimizing the perturbation forces on yaw and lateral direction, a smoother RAS can be accomplished thus aiding safety and comfort during the actual transfer. The wave force maximum is taken as 0.05685. Runs were simulated which used maximum wave forces in the range 0.1137 to 0.05685, wave lengths from 0.5 to 1.5 ship lengths and wave directions 015 to -015 degrees off the initial replenishment course. The control system handled all of the perturbations well except for the cases of a wave length of 1.5. This length of wave with a force of 0.05685 exceeded the control systems capability in that the steady state conditions were not met before a turn was commenced. Figure III-65 shows this instability in the lateral distance DY of the turn phase. It is felt that the modeling inadequacies of the sea state development of chapter II coupled with a simple adaptive gain scheme are the source of the problem. This same phenomenon is covered in greater detail in the longitudinal position offset testing portion of the velocity control section of this chapter.

Problems of this type also manifest themselves in some cases when the wave force maximum (WFMA) was close to the 0.1137 value. If the sea state becomes excessive, which this value represents, a different gain schedule or, at best, a more complex adaptive gain scheme is called for.

The plots produced by computer program #7 are presented as a representative indication of the effectiveness of the control system in the presence of a sea state. Figure III-66 gives the yaw results of the approach phase which indicates the effect of the wave action. The corresponding rudder action of figure III-67 compensates to give the smooth lateral distance shown in figure III-68. The wave profile is shown in figure III-69 with curve 2 being WY and curve 3 being WN. Curve 1 is the WX profile which was not used in this run but will be utilized in the speed control



section later in this chapter. Similar curves are portrayed for the turn phase. Figure III-70 is the yaw difference between the two ships (remembering that the reference ship is not being perturbed by the interaction forces or the wave forces). Figure III-71 is the lateral distance DY maintained by the rudder response of figure III-72. The maximum lateral separation in the turn phase is 0.0037 ship lengths (1.95 feet). The wave profile is shown in figure III-73 with the same wave force curve sequence as the approach phase.

As can be seen from these plots, the control system operates very effectively in the presence of a sea state. Again, the development of a much more complex adaptive gain scheme is required to allow exceptionally high sea state. It is felt that the control system presented in this thesis is adequate for most situations that are encountered in the RAS environment. Only the extreme perturbations that chance would allow must be accounted for in a more complex adaptive gain scheme.



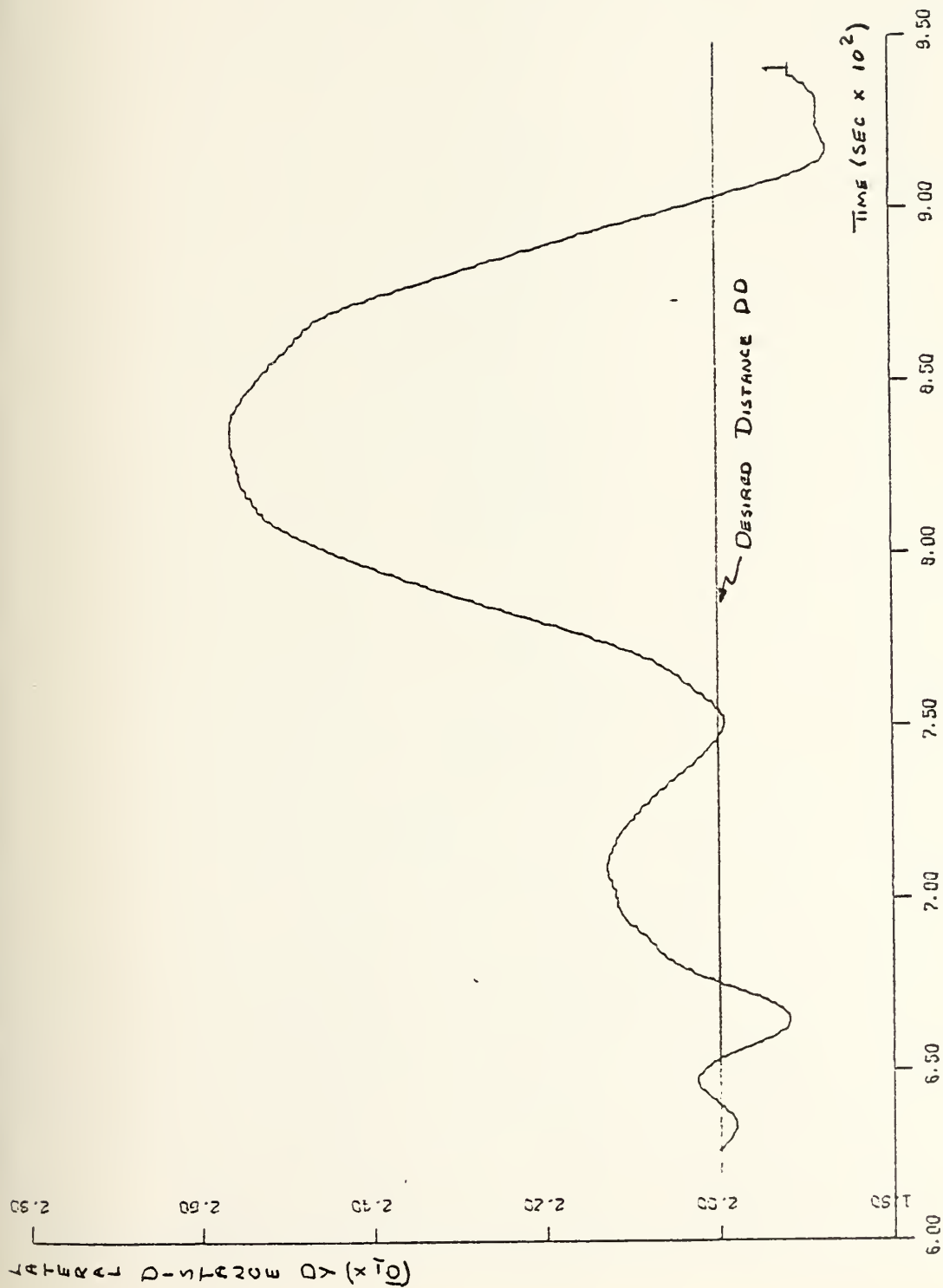


Figure III-65

Wave Effect on Turn Phase Lateral Distance (DY) WL=1.5



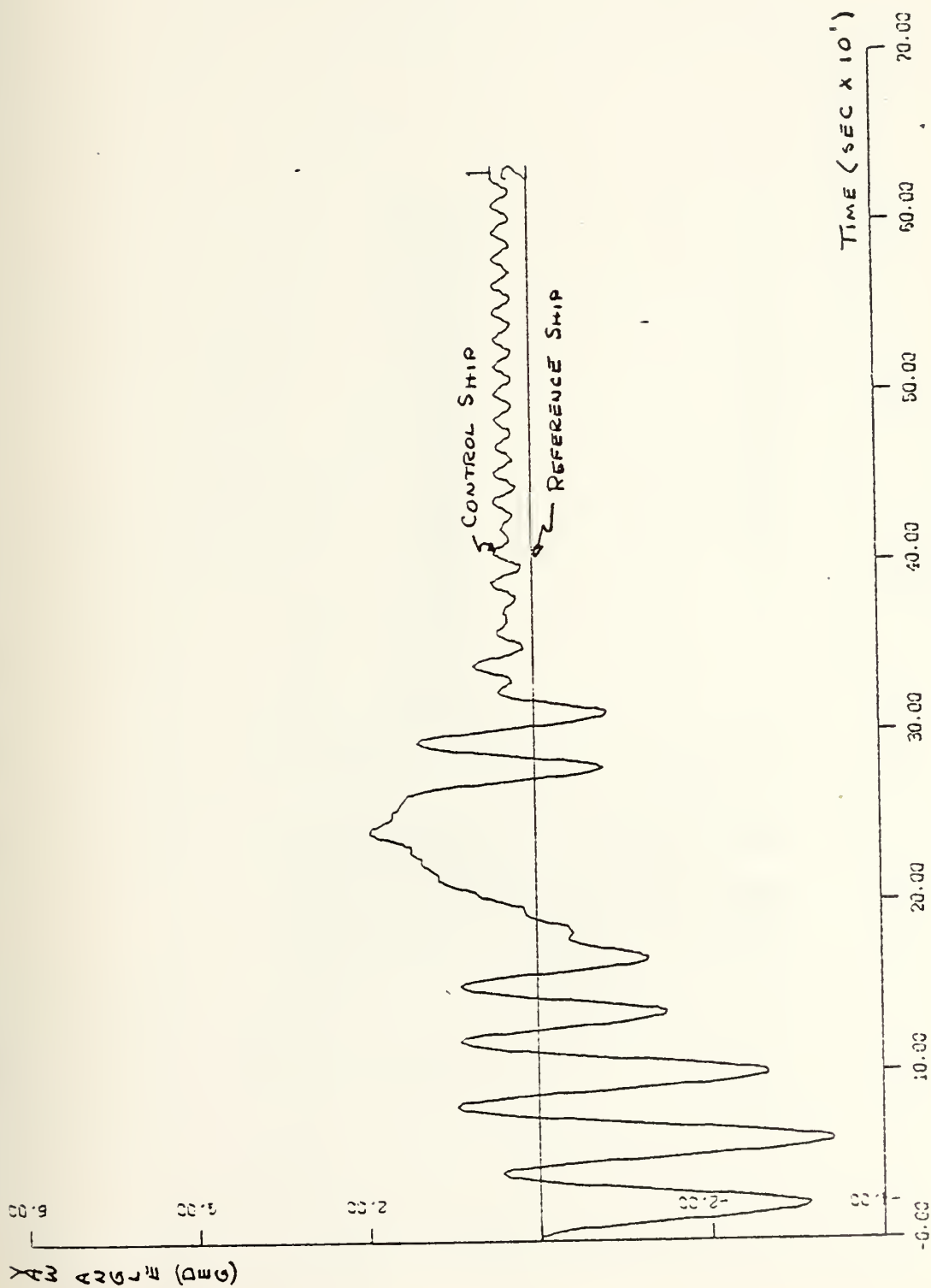


Figure III-66
Wave Effect on Approach Phase Yaw WL=1.0



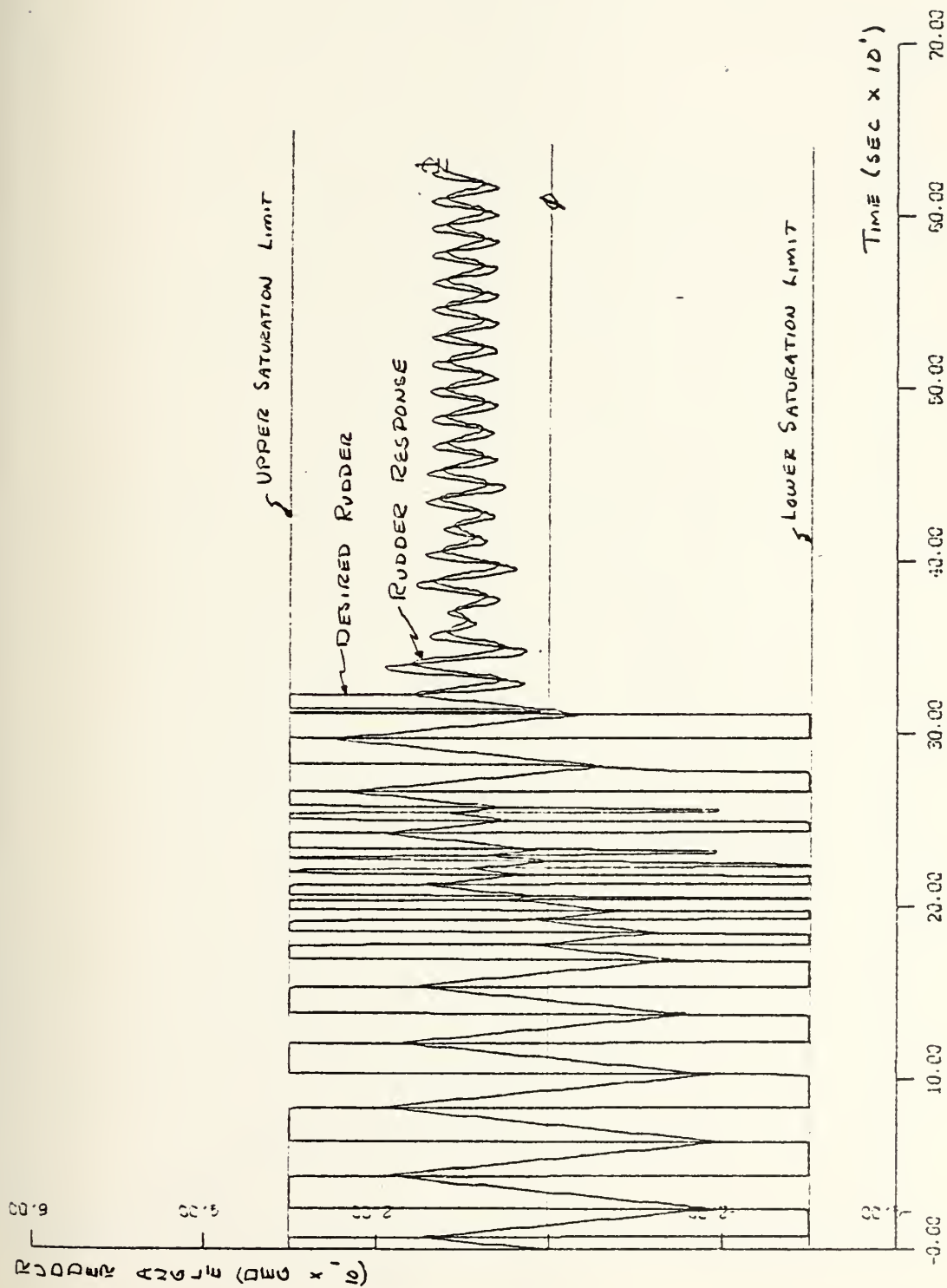


Figure III-67

Approach Phase Rudder Response to Waves WL=1.0



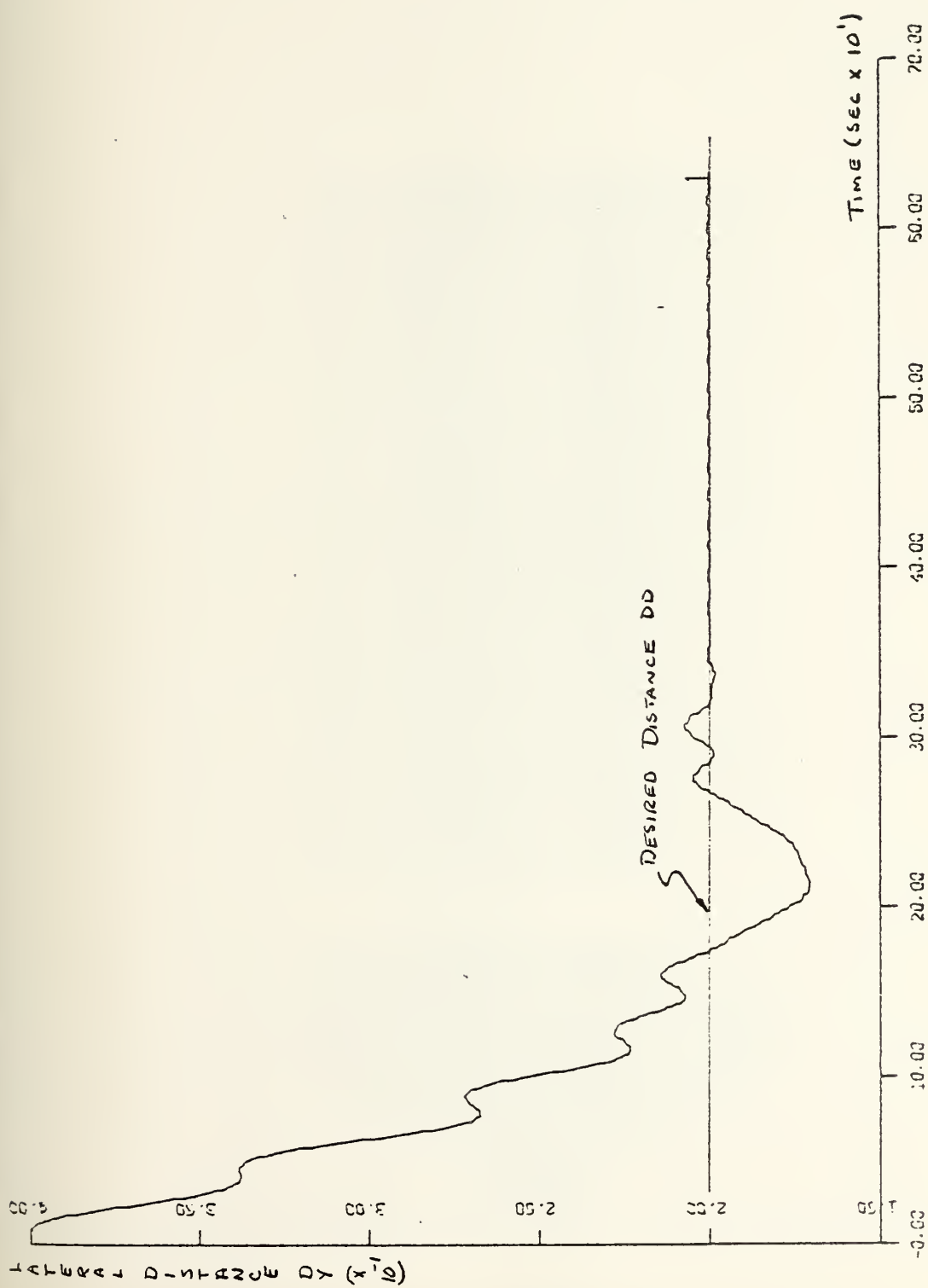


Figure III-68

Wave Effect on Approach Phase Lateral Distance (DY) $WL=1.0$



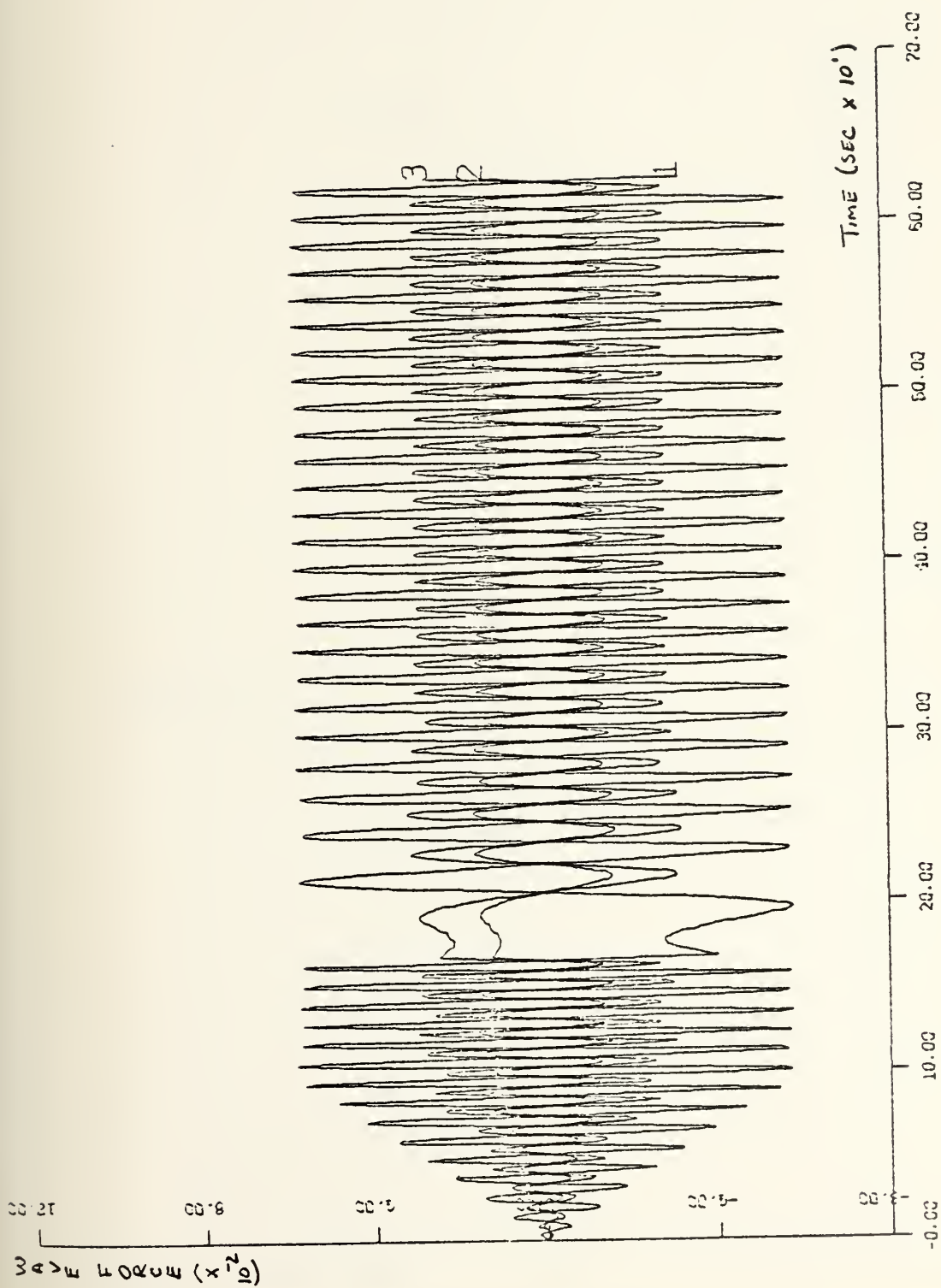


Figure III-69
Approach Phase Wave Profile WL=1.0



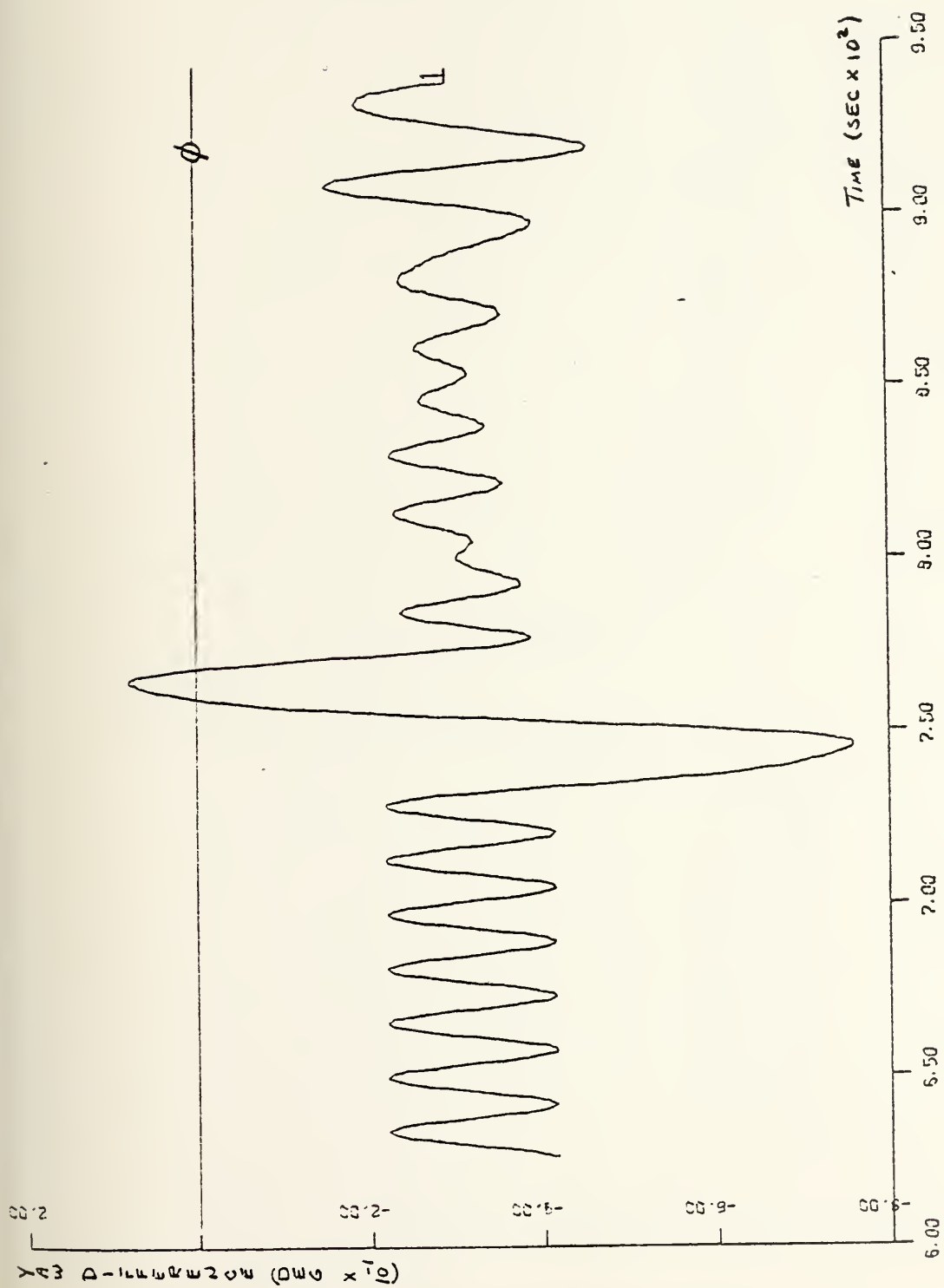


Figure III-70
Wave Effect on Turn Phase Yaw WL=1.0



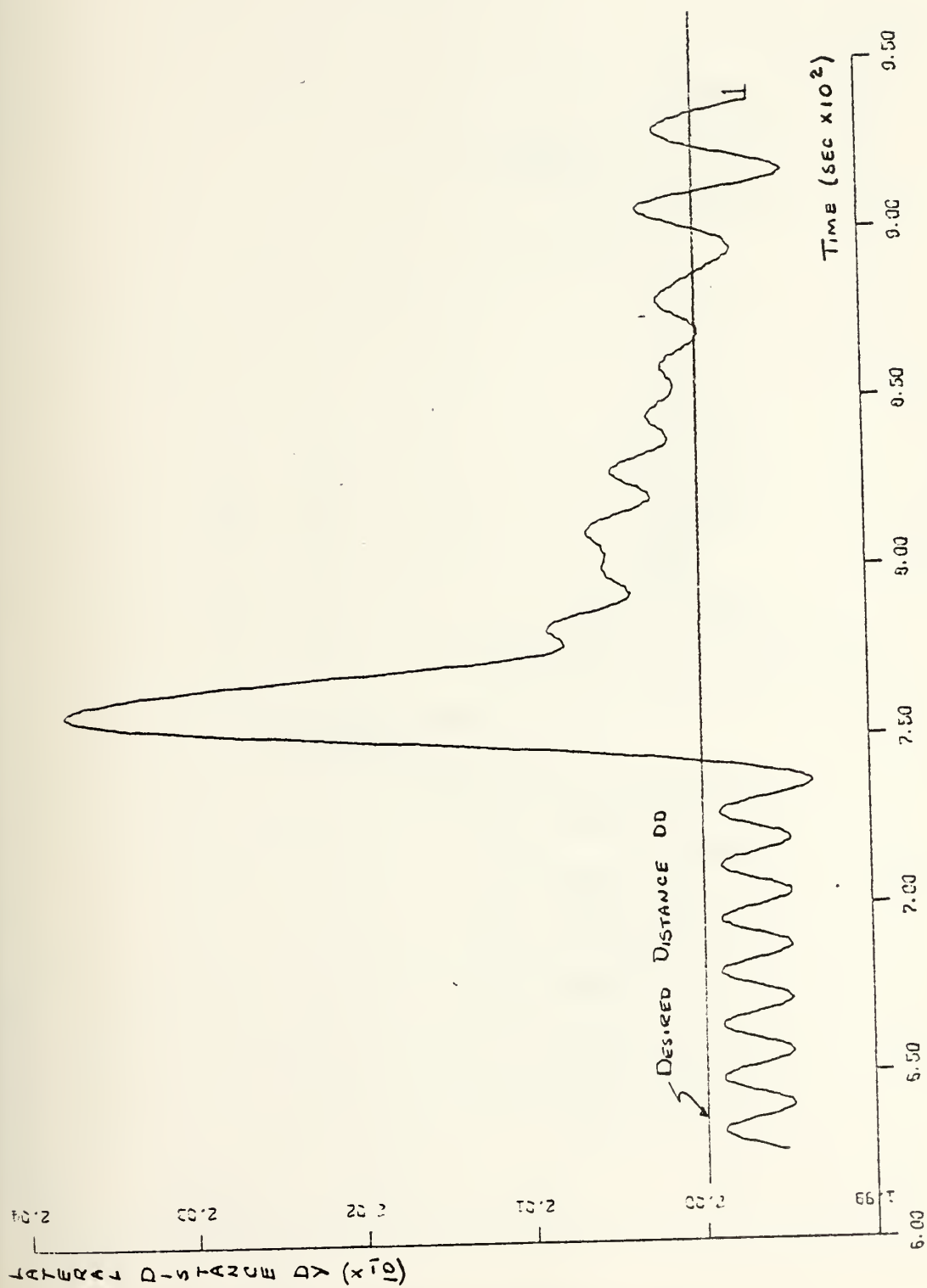


Figure III-71

Wave Effect on Turn Phase Lateral Distance (DY) WL=1.0



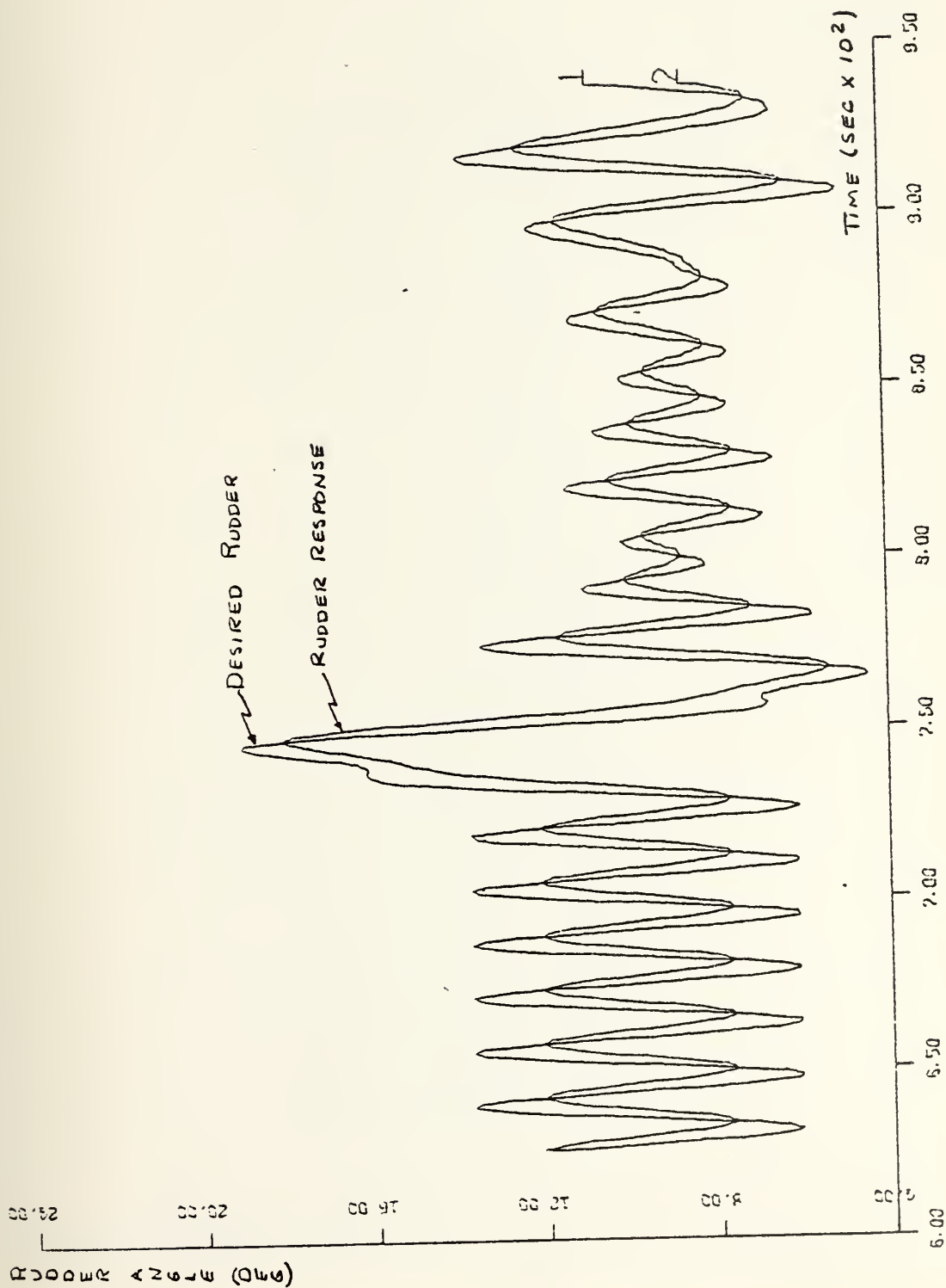


Figure III-72
Turn Phase Rudder Response to Waves WL=1.0



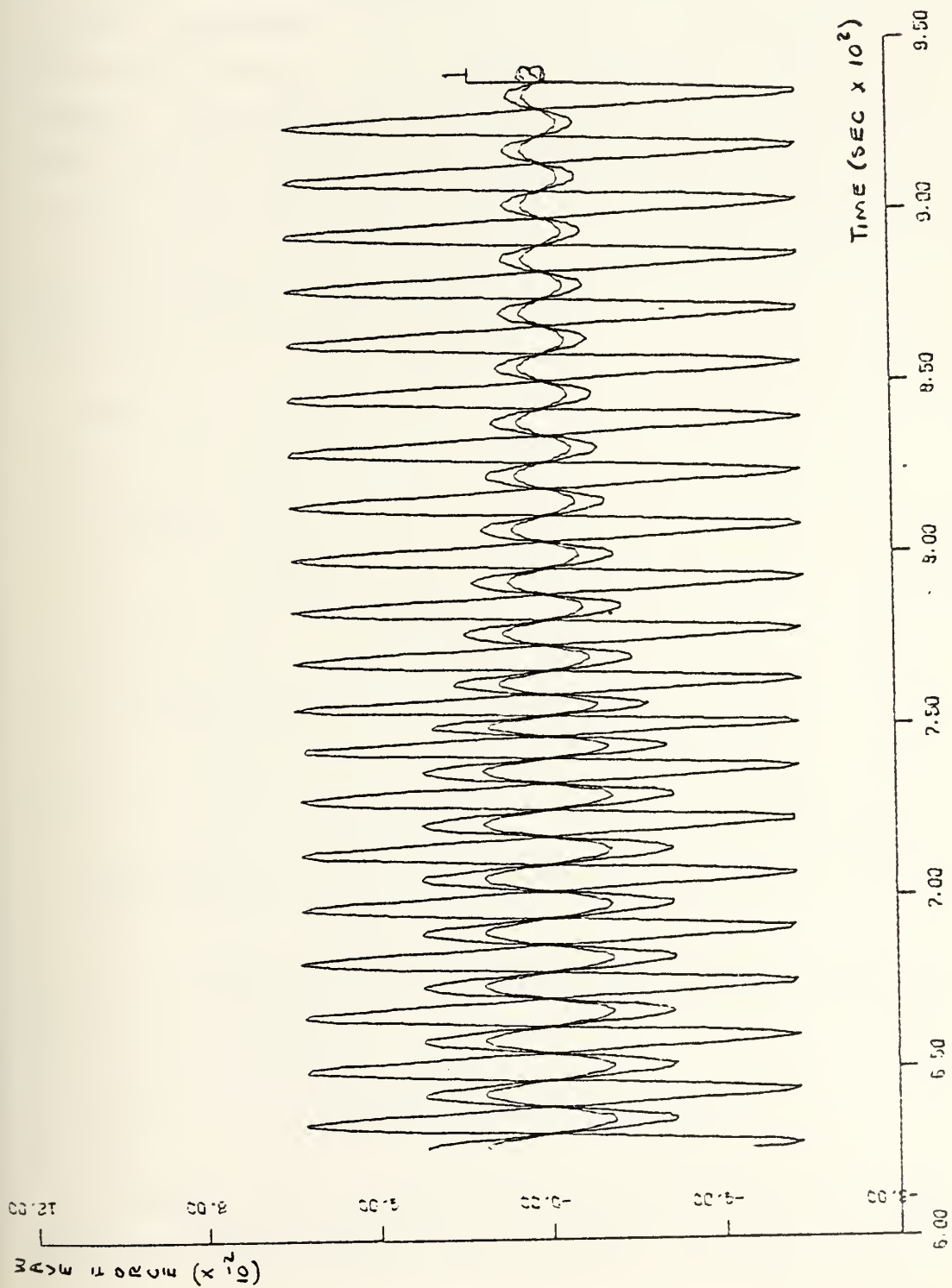


Figure III-73
Turn Phase Wave Profile WL=1.0



B. VELOCITY CONTROL

One advantage derived from using the linearized equations of motion is the decoupling of the velocity components from the remaining equations of motion. This allows separation of the design procedures for lateral separation control and velocity control. Section A of this chapter designed the lateral separation control using the simple speed control algorithm shown in figure III-74. This control output was used directly as the ship's speed (CDCT2) in the model simulation where no attempt was made to use the engine response developed in chapter II. Function SPDCTR of appendix A shows the control used.

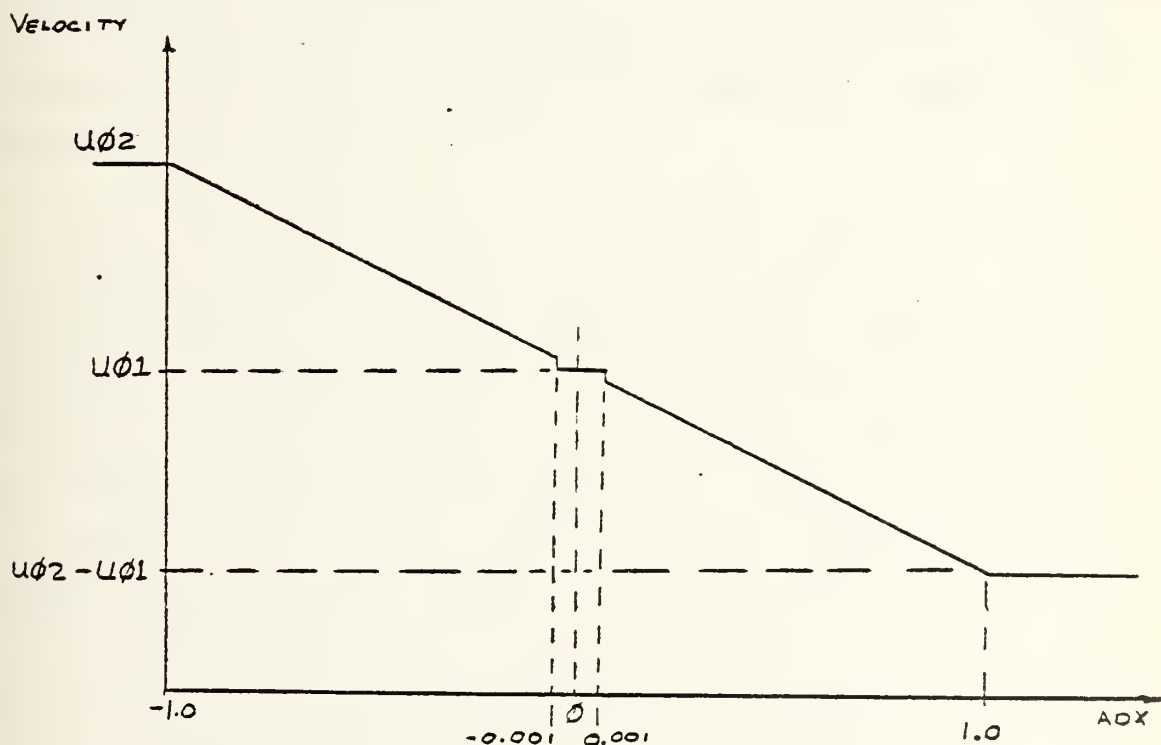


Figure III-74
Non-optimum Speed Law

Because of this decoupling assumption, any valid approach speed control can be used, if used consistently,



for such a design. However, in the RAS environment, complete disassociation is not possible. Recombination occurs in the interactive forces and moments which depend upon the longitudinal distance as well as the lateral distance. Consequently, speed, which is directly responsible for the longitudinal distance, has a direct relation to the lateral distance attainment and maintainment.

The remaining parts of this chapter deal with the development of a viable speed control algorithm and the testing of the designed system.

1. Type of Control

Whenever two ships maneuver for replenishment at sea (RAS), the prime considerations are the time required for approach and the accuracy of position keeping plus conservation of fuel.

The nonlinear control law of figure III-75 is designed to maintain a preselected approach speed for minimum approach time. The proper location of the switching point increases the complexity of the solution since the time of switching from this speed is determined by the dynamics of the nonlinear position attainment loop. Once this position is reached, the speed controller is switched down to a linear portion of the control law to allow control for perturbations about the operating position. However, small perturbations about this operating point can be tolerated and, in fact, are desired to allow for conservation of fuel. Selection of this dead zone is wholly dependent on the accuracy required for final position. Figure III-75 indicates a dead zone extending to ± 0.001 normalized distance which in this case translates to ± 0.53 feet. Systems for which fuel considerations are not a motivating



factor may be designed without this part of the control law to allow finer tracking in the position loop.

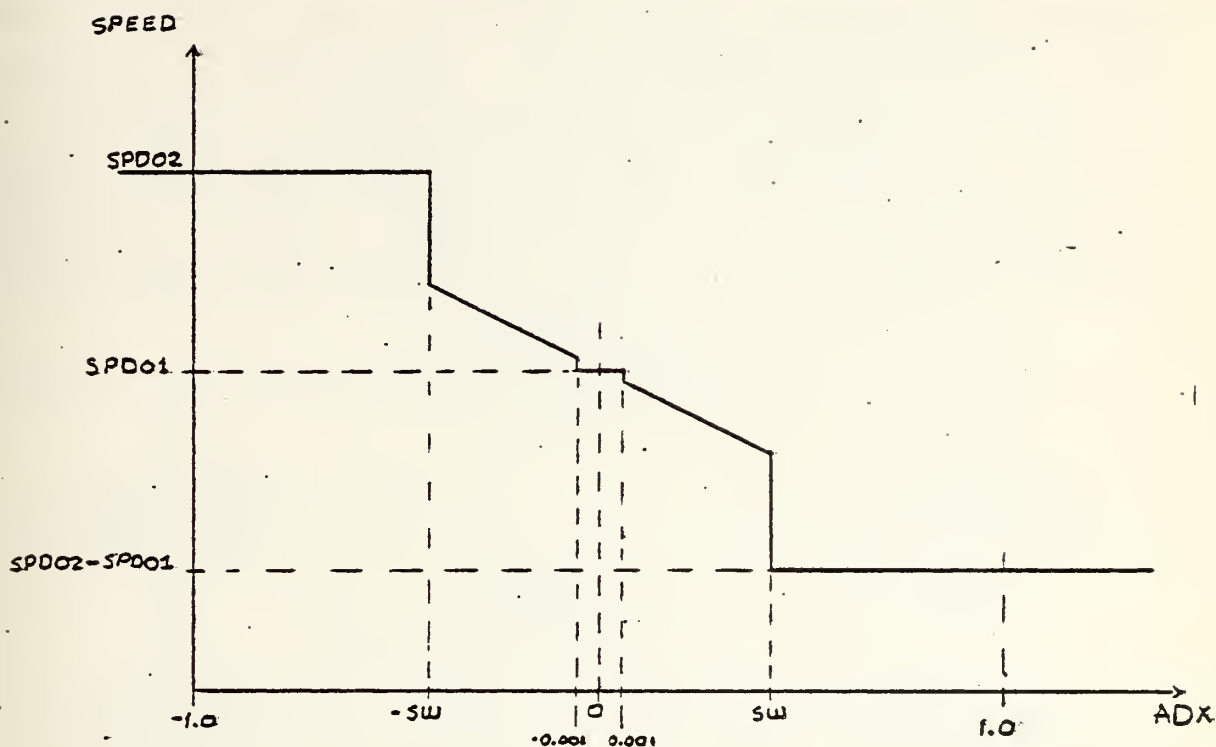


Figure III-75
Speed Control Law

The speed control law as explained above is shown in figure III-75 for an initial approach speed of SPD02 and a final estimated reference speed of SPD01, with ADX being the dynamic position feedback defined as the longitudinal distance between centers of the ships referenced to the controlled ship's heading. Analytically, the linear portion of the control law is written as:

$$SPDCTR = -ADX \cdot (SPD02 - SPD01) + SPD01$$

Symmetric continuation of the control law accounts for operation on both sides of the operating point.



2. Optimization

Using this much simplified model of chapter II and the basic control law of figure III-75, the desired switching curve can be established. An optimization subroutine such as Subroutine BOXPLX can be used to iteratively obtain the optimum switching position (SW) for representative initial approach speeds. Figure III-76 is a flow chart of the subroutines and functions required for speed control

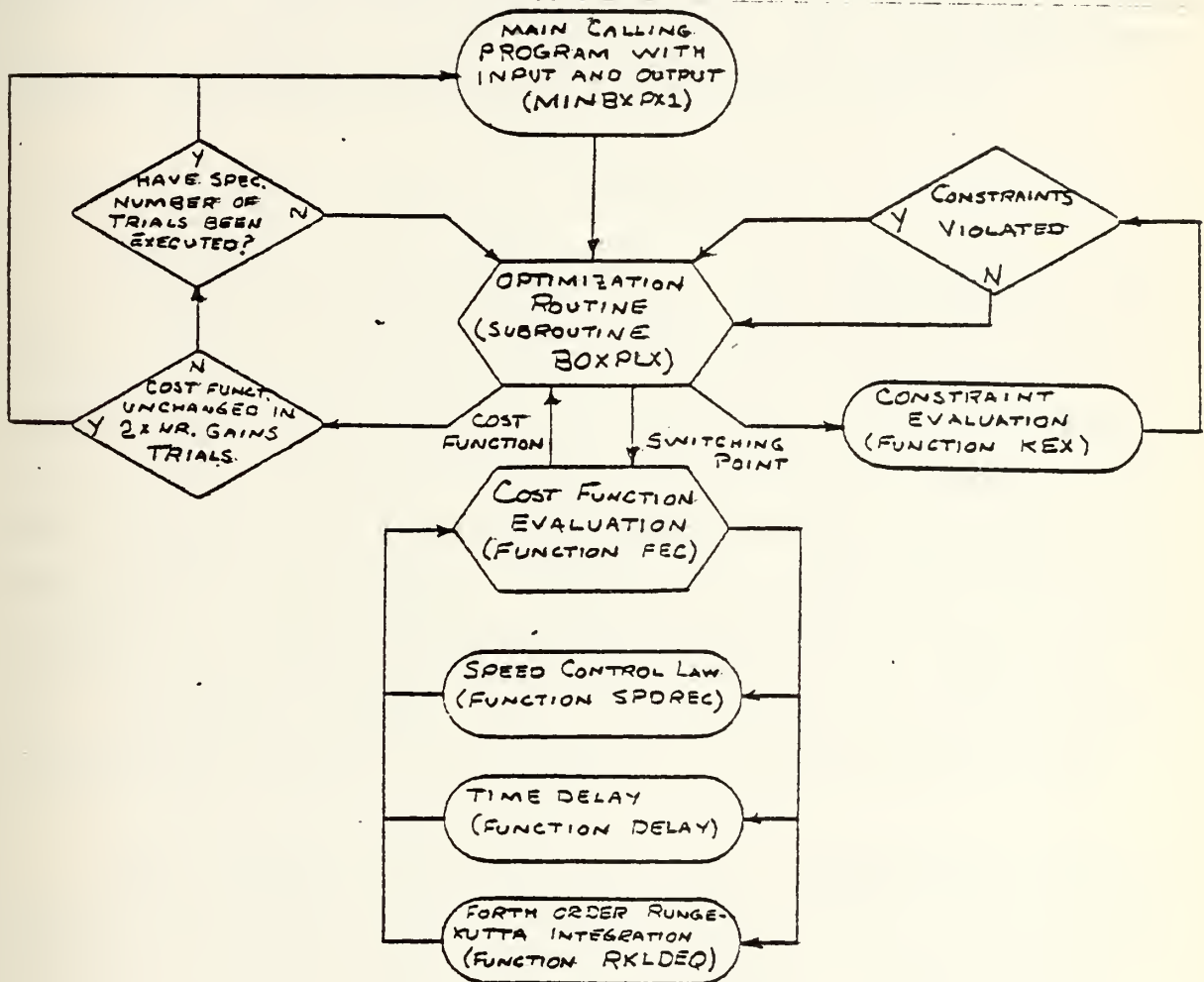


Figure III-76
Optimization Flow Chart

optimization. The major merit of this nonlinear control law stems from the predetermination of the switching point for all possible conditions of initial speeds. This apriori



knowledge allows for offline computation of the switching position prior to commencing the approach. The cost function used for optimization is the ITAE which accomplishes two objectives. First, it forces the approach to be accomplished in minimum time. Secondly, it insures that the fuel expenditure will be optimized in the elimination of most overshoot and bang-bang control in the dead zone portion of the control law. The final value of the position error must be within the specified dead zone and the terminal speed must match the reference speed (SPD01). The cost function has the following form:

$$J = \int_{t_0}^{t_f} (t \cdot |ADX|) dt$$

Table III-4 is a comparison of the optimization runs with various initial speeds. The values shown for SW must be multiplied by the speed differential (SPD02-SPD01) to obtain the corresponding value of ADX. The max/min values show the band of values which produce the optimum cost. This range of values is attributed to the integration step size used in the optimization program. Experience with this particular optimization program indicates that erroneous values of the switching point are found if the step size is not carefully chosen. The step size may be adequate for integration, but not for location of the switching point.

The points obtained from the optimization runs are plotted in figure III-77. These points define the nonlinear switching curve which must be stored in the computer to insure optimal operation of the speed control for all approach speeds. From here there are many procedure options open. These options have as a goal some usable form for predicting the optimal switching point for any set of initial conditions. One may choose linear straight line segments with an interpolation routine, or a closed form switching curve polynomial. Due to the availability of a



INITIAL CURVE POINTS					
SPDO2	SPDO1	SW MAX	SW MIN	SW	COST
1.1	1.0	.545 *	.545 *	.545	22.340515 **
1.2		.58705	.58424	.585	5.733367
1.3		.62656	.6256	.626	2.700768
1.4		.6845	.68234	.683	1.672599
1.5		.73169	.7283	.729	1.223071
1.6		.7644	.76142	.763	0.992283
1.7		.7945	.7926	.7936	0.861552
1.8		.82178	.81945	.82	0.774621
1.9		.8501	.8439	.85	0.757244
2.0		.8673	.86375	.865	0.730168
CURVE CHECK POINTS					
1.5	1.1	.6859	.6823	.683	1.668055
1.5	1.2	.6307	.6297	.6302	2.691659
1.6	1.2	.67965	.67906	.6793	1.659956

* cpu usage over 4 min. - run not complete.

** cost function based on 20 min problem time
all others based on 10 min problem time.

Table III-4
Optimization Results



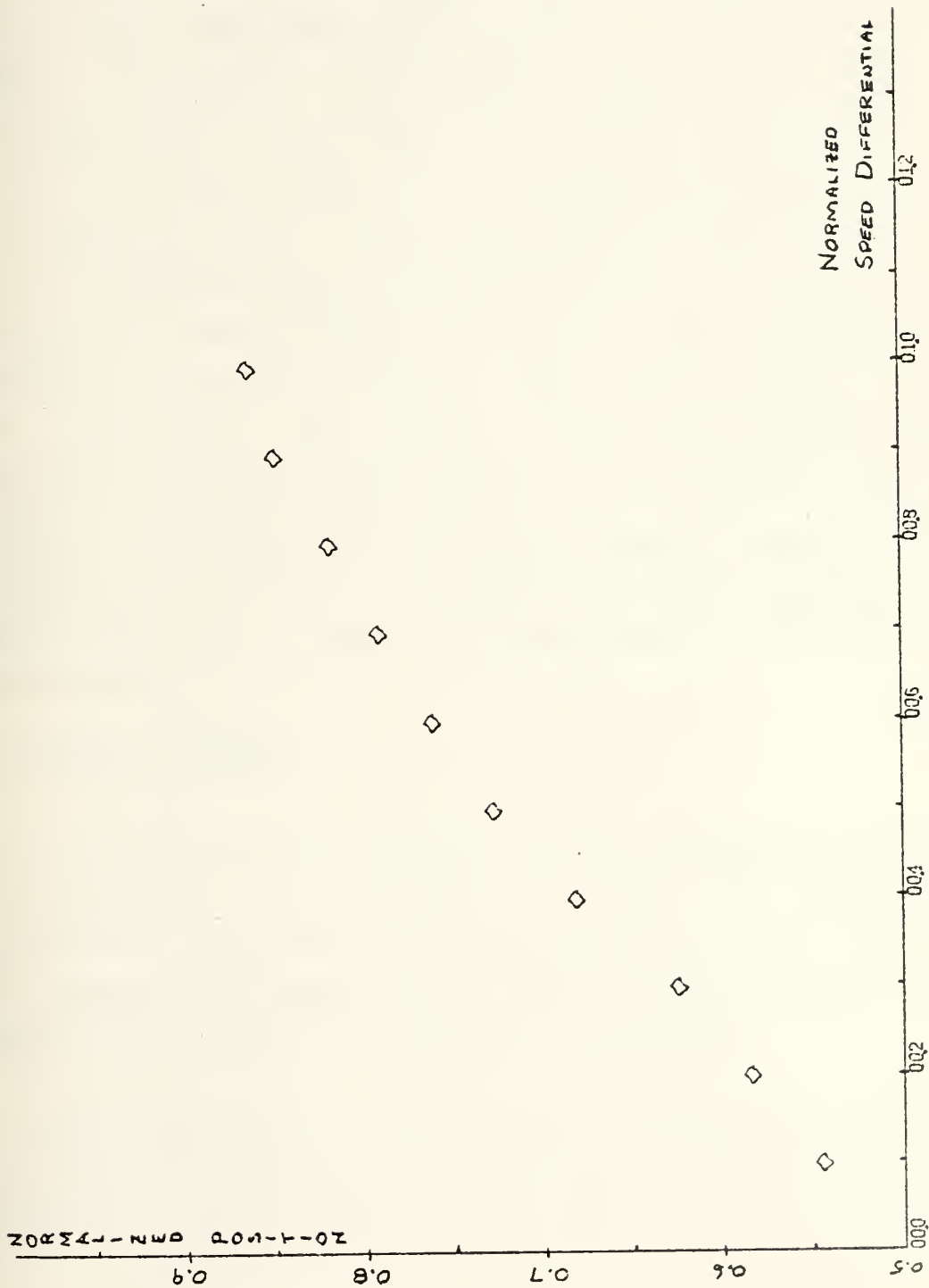


Figure III-77
Switching Curve Minimization Results

hybrid configured XDS 9300 digital computer and AGT-10 graphics terminal, the latter course was taken.

A polynomial curve fitting algorithm was used to obtain the required polynomial coefficients of best fit. This was done for polynomials of order 1 thru 5. The coefficients and the sum of the squares of deviation from the original points are tabulated in table III-5. The selection of the order to be used is highly dependent on the degree of accuracy required. In the RAS problem, the average error introduced for a first order fit is 8.0 feet(1.07 sec), while the fifth order fit introduces an average error of 1.35 feet(0.180 sec). Prior acceptance of errors introduced by an integration (and problem) step size of 0.8 sec allows for use of a second order fit without any degradation of simulation accuracy [second order average error is 2.848 feet (0.38 sec)]. The graphic display of figure III-78 indicates very little difference in the switching curves for second to fifth order polynomial fits. For the sake of accuracy, and owing to the computer control methods of this thesis, the fifth order polynomial fit shown separately in figure III-79 is used for determination of the switching point location.

3. Control Testing

A true test of the control law is accomplished when it is introduced in a computer program for a complete RAS simulation. Considering the performance of this controller in a complex environment of full scale RAS simulation allows maximum verification of the controller design.

The scenario for this simulation initially positions the ships such that the ship being controlled starts an approach 5 ship lengths (2639 feet) behind the reference



POLYNOMIAL DEGREE	COEFFICIENTS OF POWER:						SUM OF SQUARES OF DEVIATION
	5	4	3	2	1	0	
1					0.367928	0.524	2.2911×10^{-3}
2				-0.194621	0.582011	0.481	2.9117×10^{-4}
3			-0.174164	0.0927511	0.449472	0.496	1.9748×10^{-4}
4		0.748543	-1.82096	1.29791	0.120113	0.521	1.0517×10^{-4}
5	-2.24869	6.93243	-8.04233	4.08065	-0.409977	0.554	6.5732×10^{-5}

Table III-5
Polynomial Curve Fit Results



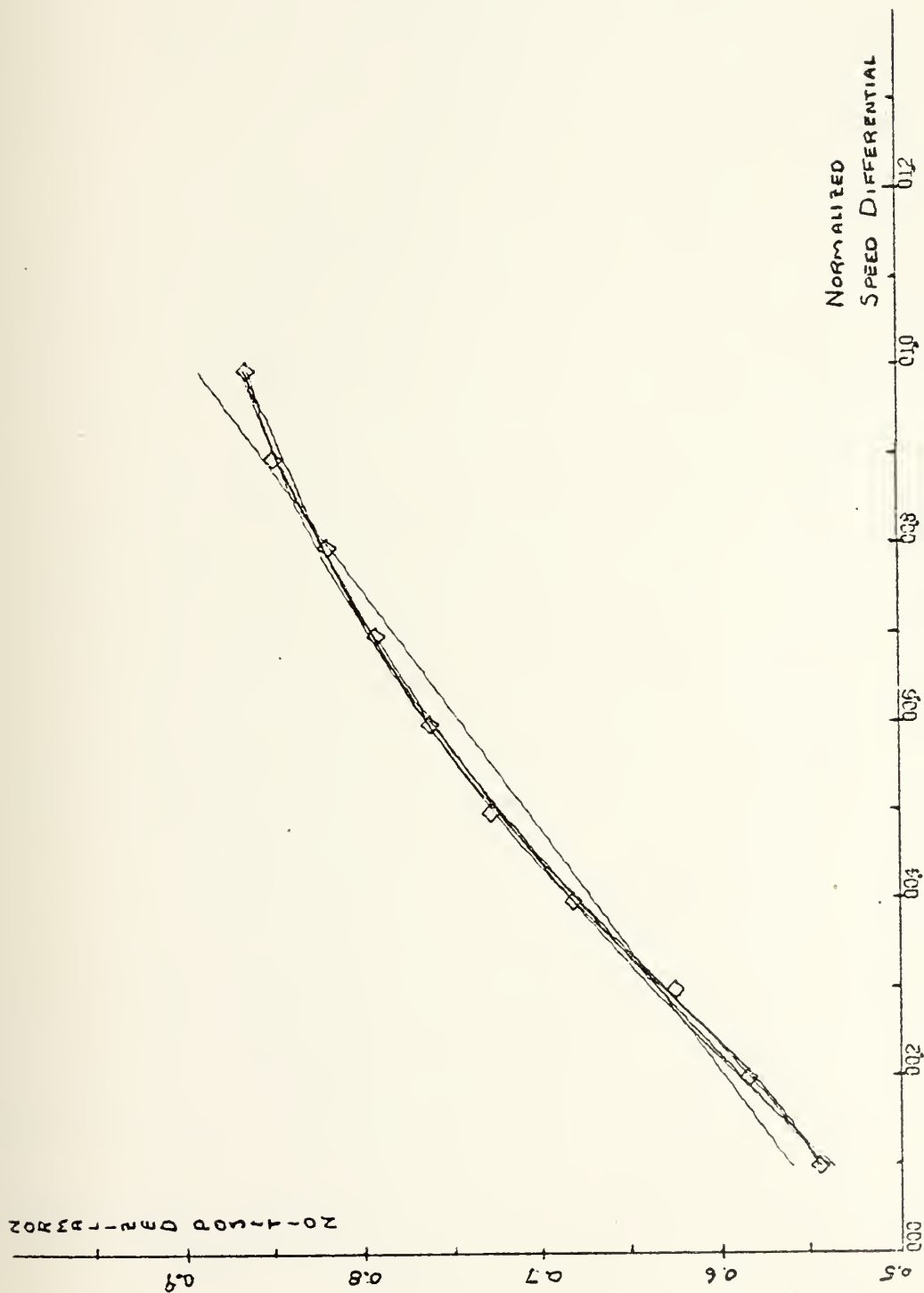


Figure III-78
First Thru Fifth Order Polynomial Curve Fit Results



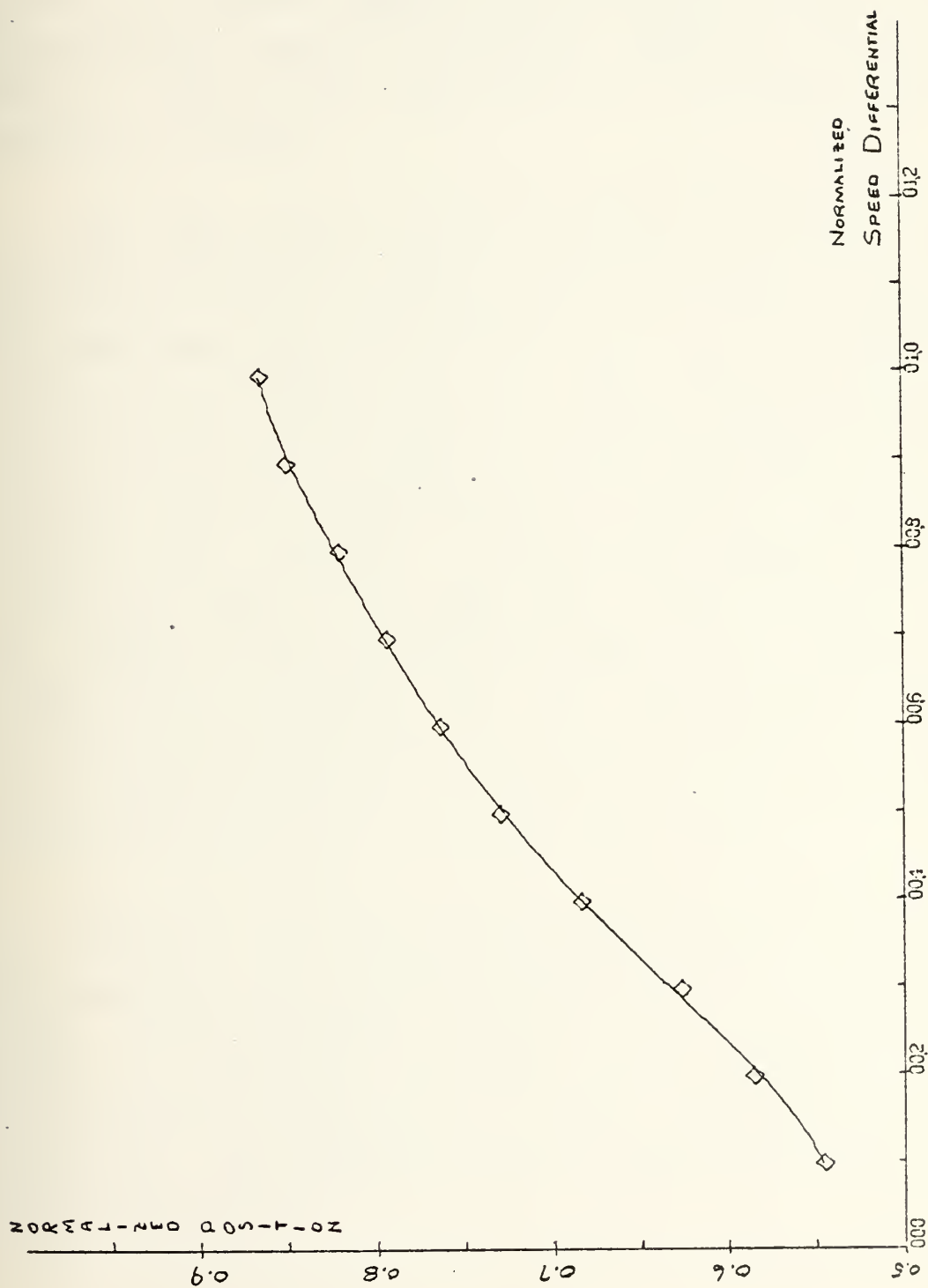


Figure III-79
Fifth Order Polynomial Curve Fit



ship and displaced 0.4 ship lengths (211 feet) to the right. The desired final position is alongside and displaced 0.2 ship lengths (106 feet). The heading control system used is developed in section A of this chapter.

The approach phase is accomplished with the speed desired and speed acquired shown in figure III-80 with the corresponding position attainment exhibited in figure III-81. These plots show excellent switching and optimal position attainment.

The next step is to insure that the position keeping loop will maintain the desired position with an induced perturbation. This is accomplished by turning the reference ship away from the control ship a total of 15 degrees to observe the reaction of the speed control loop. The reference ship's turn causes the relative motion between the ships to be altered, making the control ship lag the desired position. The nonlinear control system is designed to correct this situation as soon as the actual position is outside the limits of the dead zone. Figure III-82 displays the desired speed and acquired speed for the control ship. Figure III-83 indicates that the corresponding position deviates from the desired by 0.0154 ship lengths (8.13 feet) at the maximum excursion. This is well within the limits of acceptability for such a drastic perturbation.

The introduction of velocity control was accomplished by combining the simplified engine response of chapter II and the speed control law developed here. By setting the speed desired (SPDDES) equal to the output of Function SPDREC and scaling the speed error (SPDERR) to the nondimensional equations of motion, the velocity loop is initiated. The auxiliary equations added to those presented in chapter II are:



Figure III-80
 RAS Speed Control Approach Phase
 Speed Desired (1) and Speed Acquired (2) vs Real Time

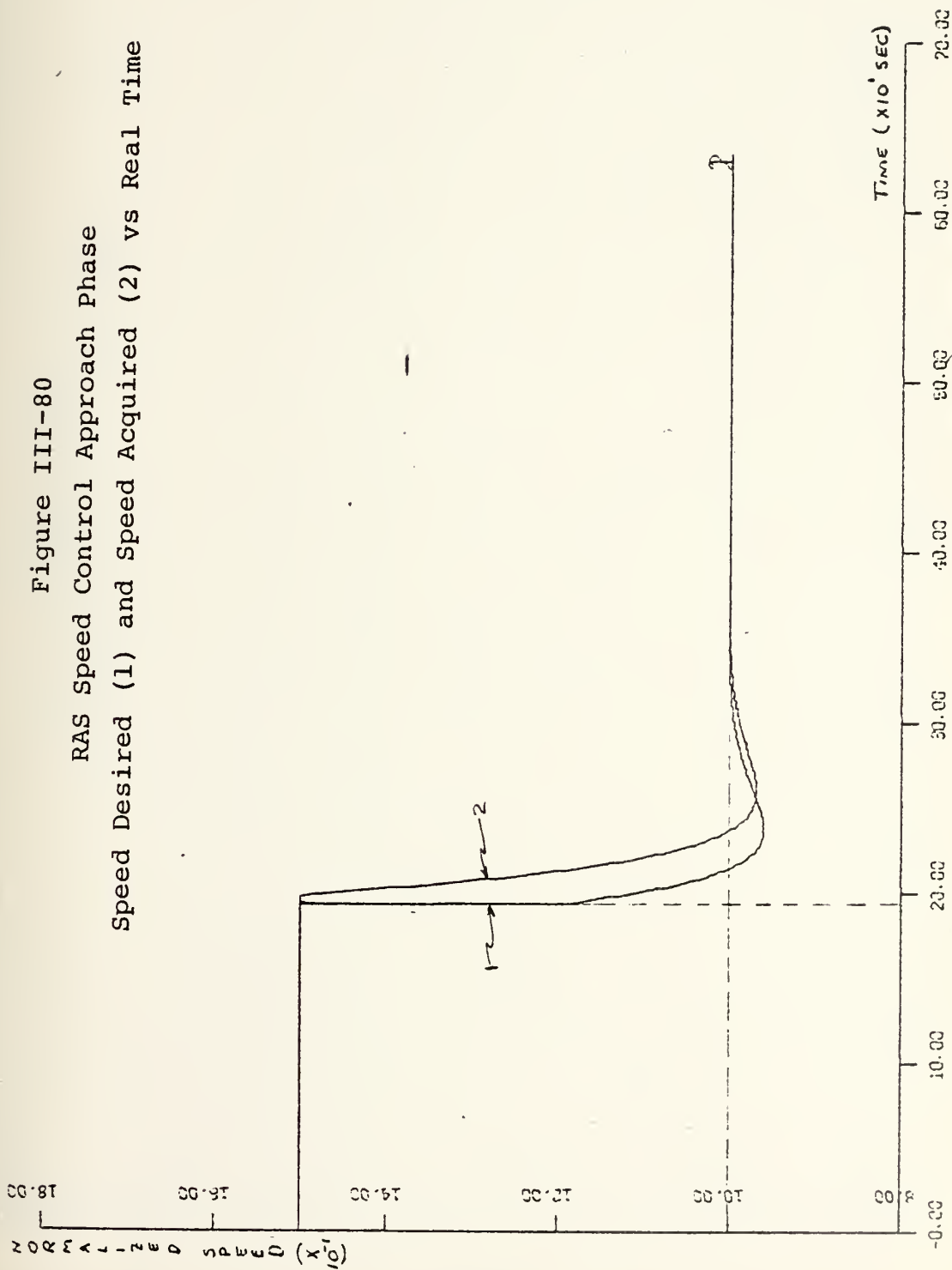




Figure III-81
 RAS Speed Control Approach Phase
 Position Attainment vs Real Time

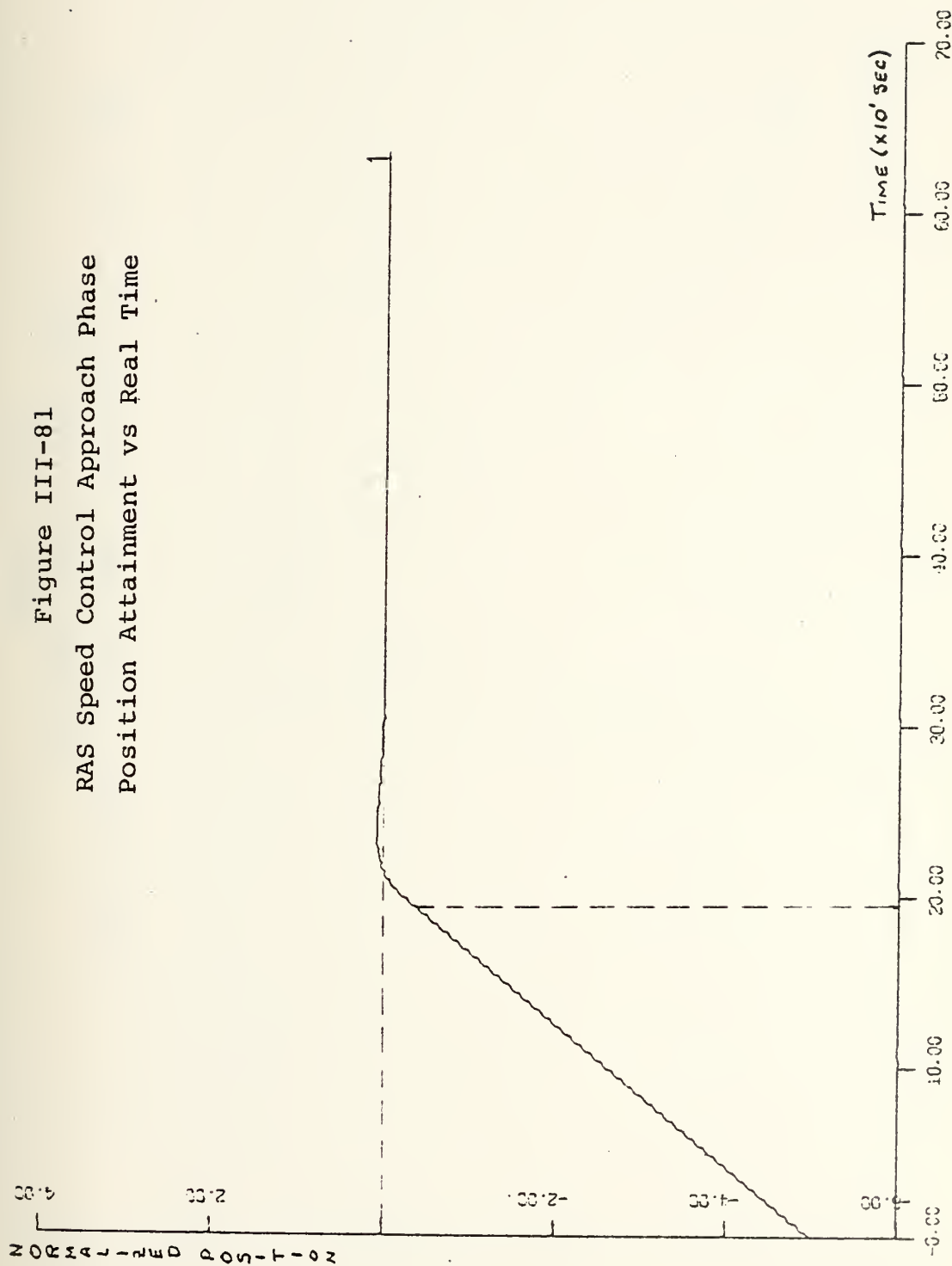




Figure III-82
 RAS Speed Control Turn Phase
 Speed Desired (1) and Speed Acquired (2) vs Real Time

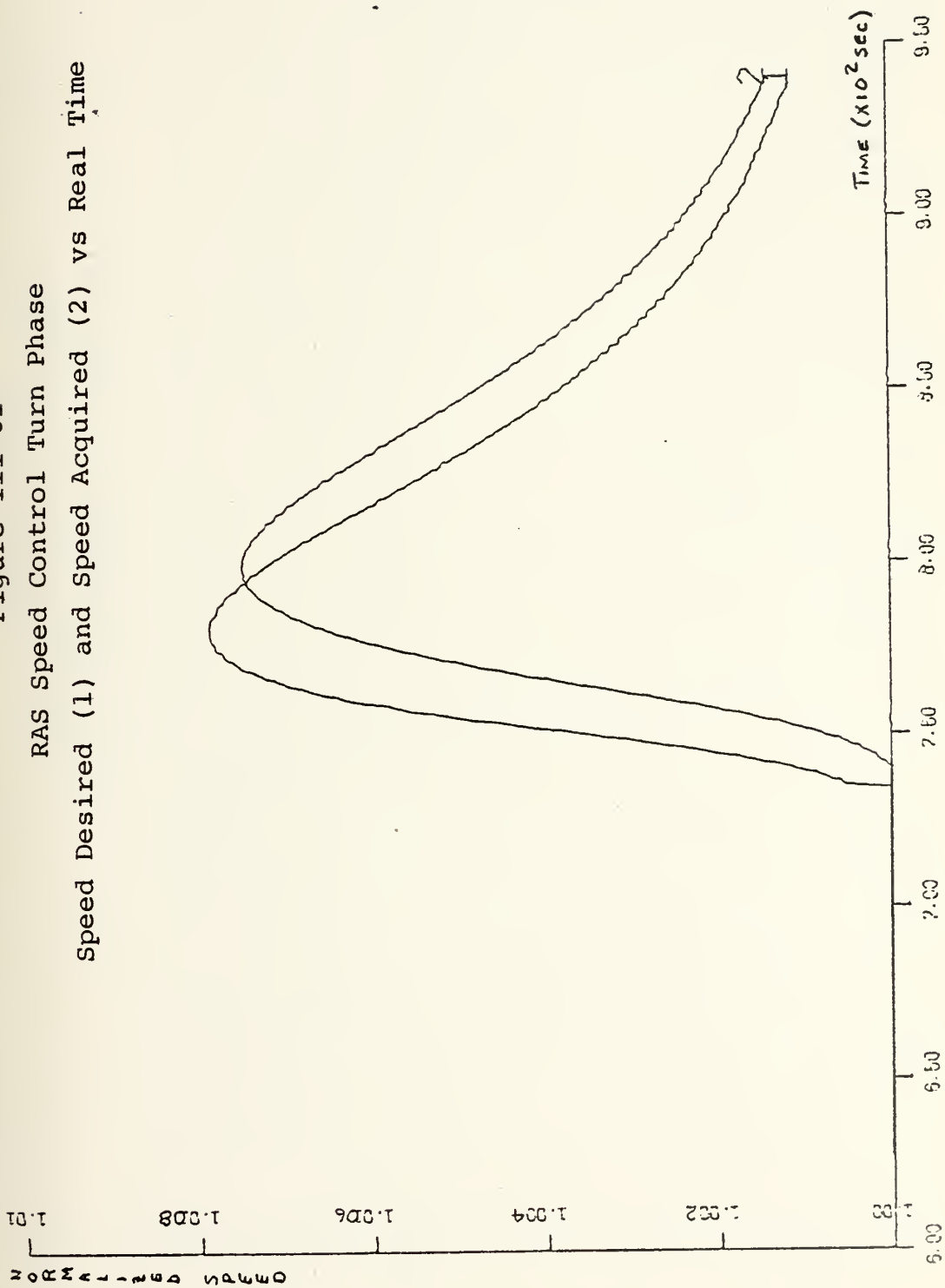
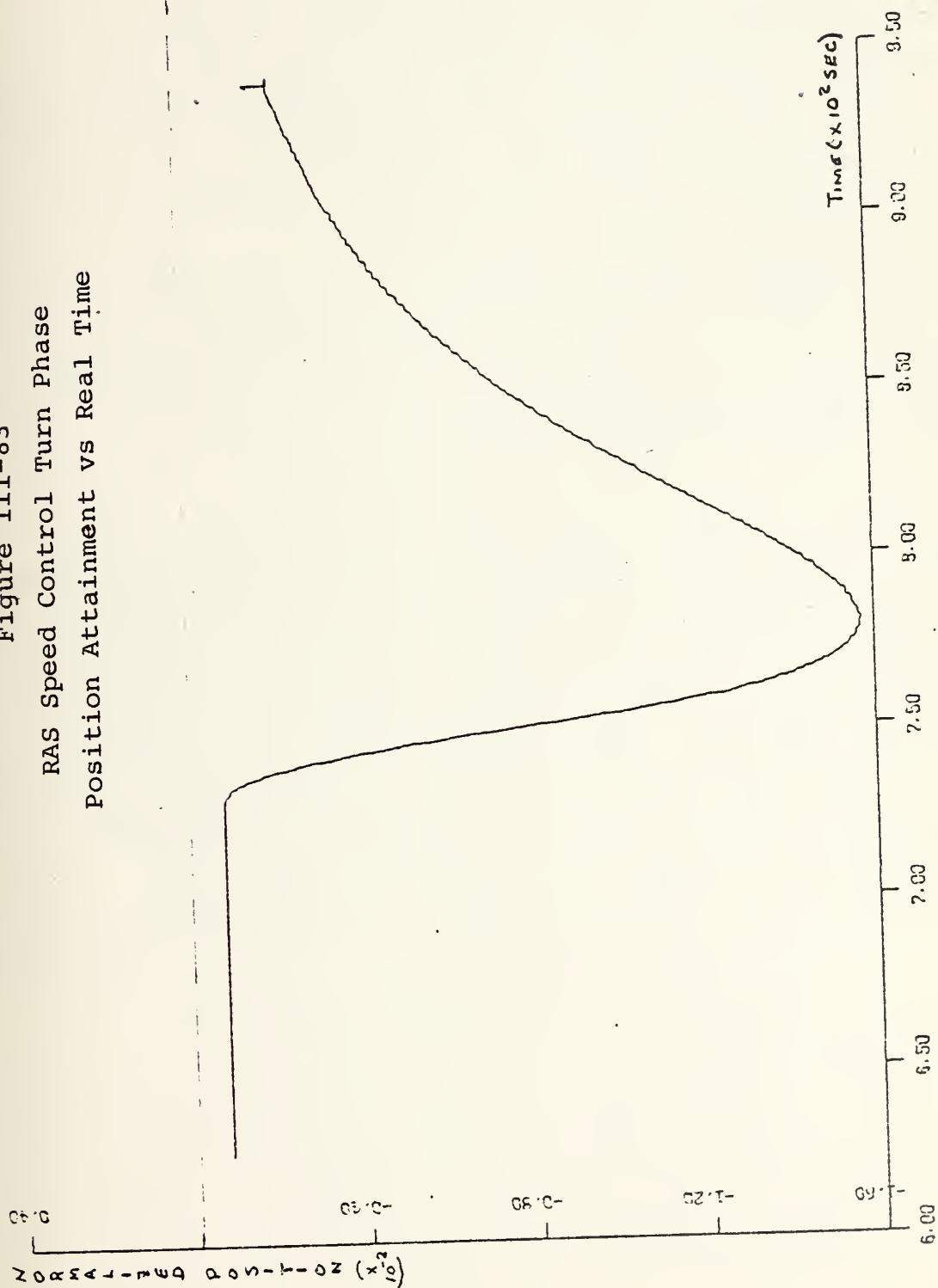


Figure III-83
 RAS Speed Control Turn Phase
 Position Attainment vs Real Time





```
SPIDES = SPDREC (ADX,SPDO1,SPDO2,SW)
CDCT2 = INTGRL(U02,SPDERR*LUC)
```

These equations are introduced in computer program #8 to produce figures III-80 thru III-83.

Further system study indicates that the reference ship speed must be known to a fairly high degree of accuracy. Without apriori knowlege of the reference ship speed, a constant bias is introduced. The amount of bias allowable defines the permissible uncertainty in the reference ship's initial speed. This bias can amount to as much as 0.1 ship lengths (84.48 feet) for a reference speed inaccuracy of 2.5 kncts (0.1 normalized speed). However, it is felt that the reference speed in any practical situation will initially be known to within 0.5 knots (0.02 normalized speed). This more practical error will introduce a bias of only 16 feet.

Other feedback parameters can be used to offset the lack of apriori knowlege of the reference ship speed. Since the reference ship is tracked with a high accuracy range and bearing device and the controlled ship's speed is measured, a decoupled multivariable scheme is used to further refine the reference ship speed. With high resolution devices presently available^[14], it is estimated that this can be done practically to within 0.05 knots (0.002 normalized speed). This would bring the offset bias to 1.6 feet; well within previously defined errors introduced by integration step size.



4. Longitudinal Position Offset

Throughout the development of the heading control and speed control, the scenario has followed the condition that the final position would be longitudinally alongside. Although this is a good assumption for ships of the same type, it does not account for RAS station differences for different ship types. To alleviate this disparity, function SPDREC was redesigned to allow pre-planned offset condition to exist. Function SPDOFC of appendix A is a result of this redesign.

Simulation runs, with a change of the speed control function only, resulted in some unstable conditions existing in the heading control loop. The cause of this phenomenon stems back to the adaptive gain scheme used and the changes made to force the control loop to a steady state value prior to a turn. By using a favorite ploy of experienced conning officers, this problem is alleviated. The ploy is to take the ship alongside and then either drop back to station or surge forward to station. This method is accomplished by setting the initial offset (XOPS) to 0.0. The final desired offset (XOFSD) is stored and not used until the ship is settled out alongside. It is subsequently used as shown in the following Fortran code:

```
IF (ATIME.GT.450.0) XOPS = XOFSD
```

This method solved the gain transition problem. It did not, however, give a completely stable simulation run. Unstable conditions still existed at the end of the turn phase. This is not surprising, considering the heading control optimization method used. The set of gains previously found were for the alongside scenario only.

Different interactive forces and moments at the offset position cause these gains to be no longer optimal.

By relaxing the control loop in the heading velocity feedback gain (VFBG), sub-optimal control at all practical offset positions is achieved. The gain VFBG was changed from 0.084028 to 0.1 in the turn phase adaptive gain schedule without significant loss of control efficiency for alongside operation (2.3 feet maximum excursion vice 2.0 feet previously obtained). Subroutine SWTCHF of appendix A reflects the gain change and offset calculations required. Computer program #9 incorporates the changes required for offset simulation. Table III-6 is a cross reference listing of the plots obtained. From these figures, the effect of different longitudinal positions is readily apparent. An offset of 0.1, equating to 52.8 feet, causes greater lateral excursions when astern (XOFSD = -0.1) of the alongside position than when ahead (XOFSD = 0.1). The longitudinal position maintainment, however, is essentially the same in all cases.

Run	Approach Phase Plots			Turn Phase Plots		
	A	B	C	A	B	C
XOFSD	0.0	0.1	-0.1	0.0	0.1	-0.1
Lateral Distance DY	84	86	90	94	96	99
Yaw Difference	85	87	91	95	97	100
Speed Response	80*	88	92	82*	82*	82*
Longitudinal Position DX	81*	89	93	83*	98	101

* Note: These plots are the same as those obtained from computer program #8 and are not repeated here.

Table III-6
Position Offset Testing Cross Reference



An alternative to the method shown here is again a completely adaptive gain scheme which would achieve optimal control instead of the sub-optimal control settled for here. The alternative may become even more important if the nonlinear terms of the equations of motion are considered. This would couple the heading and speed control designs to a larger extent than encountered in the interactive forces and moments.

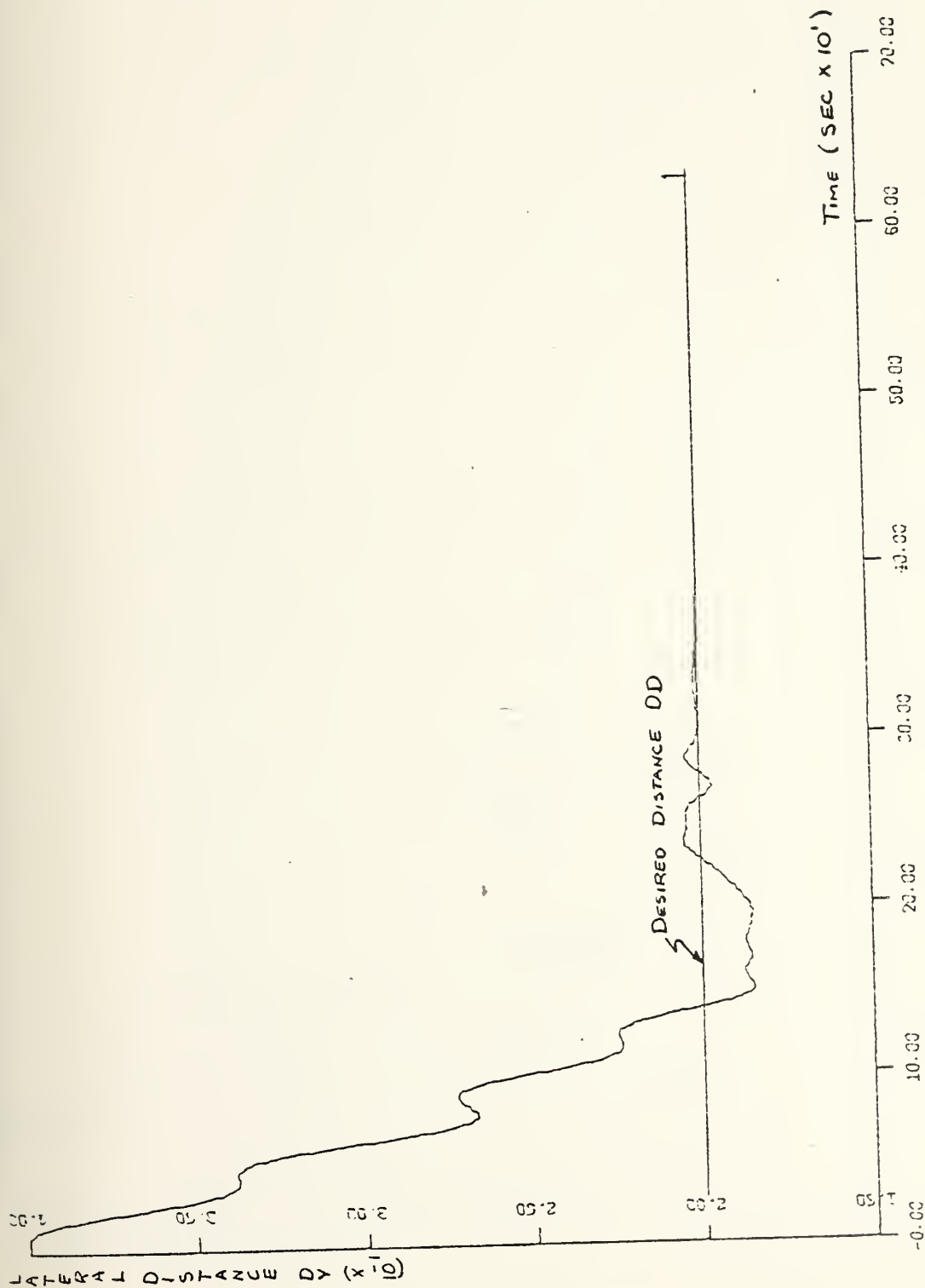


Figure III-84
Approach Phase Run A Lateral Distance DY



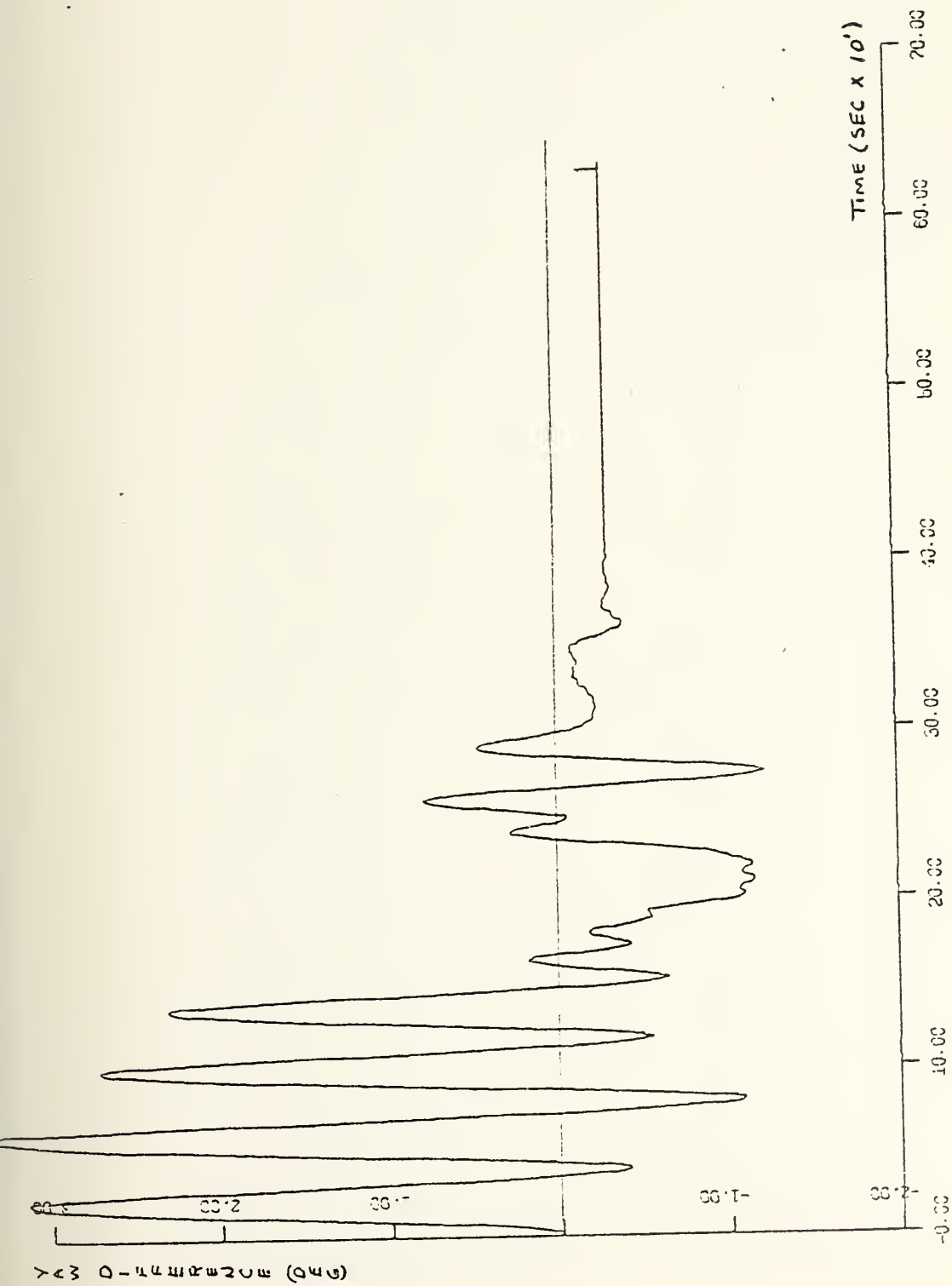


Figure III-85
Approach Phase Run A Yaw Difference

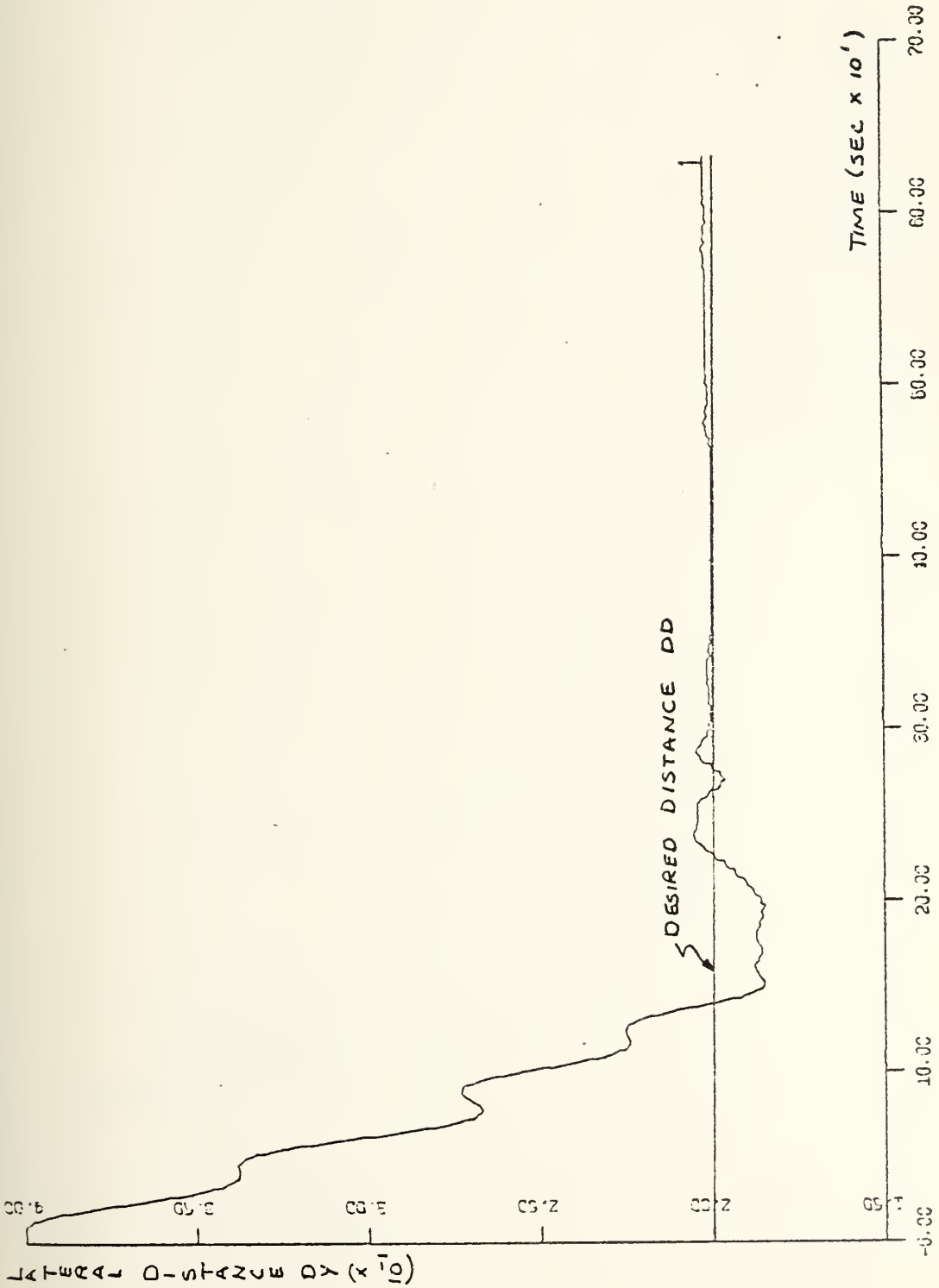


Figure III-86
Approach Phase Run B Lateral Distance DY

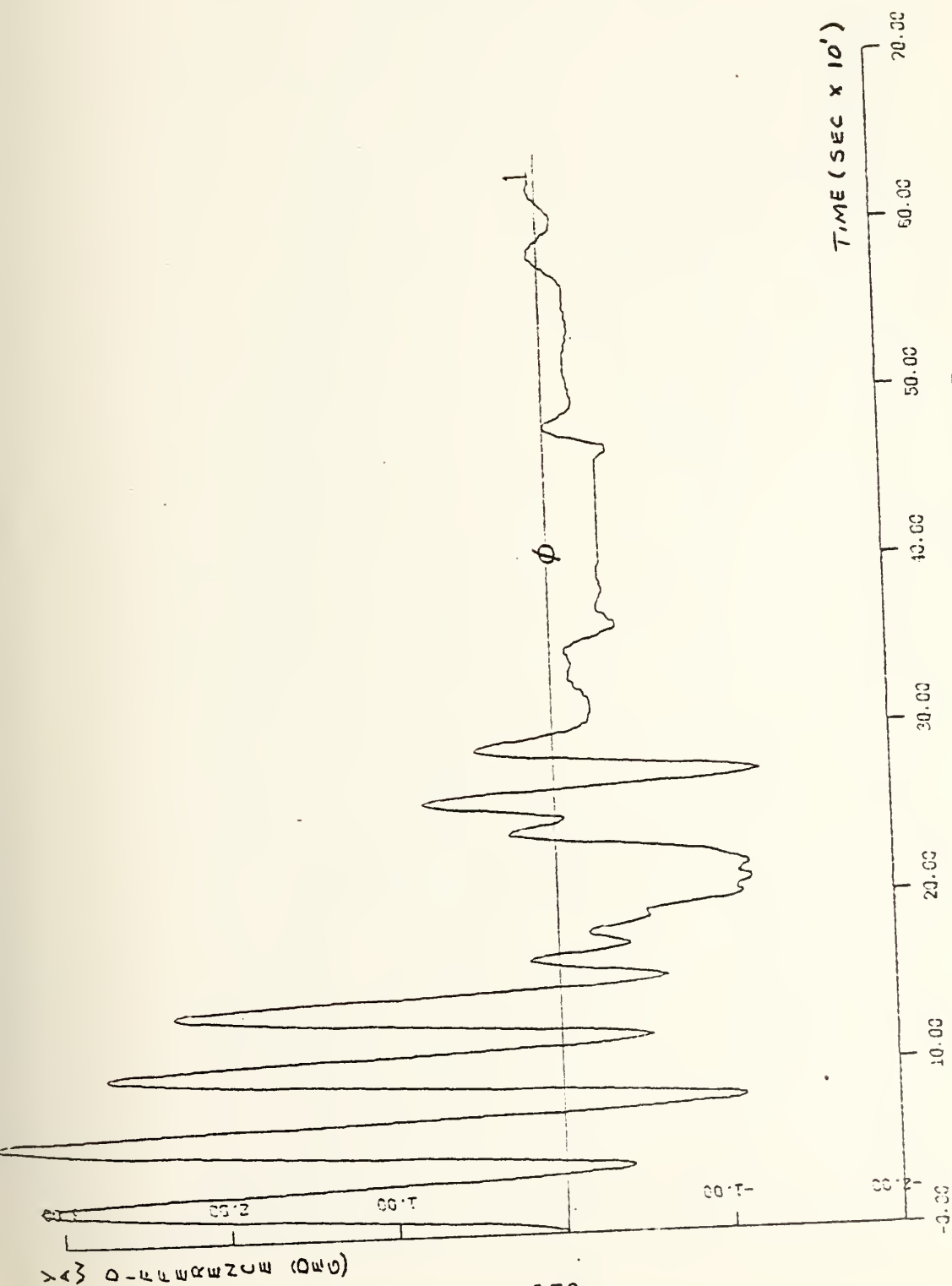


Figure III-87
Approach Phase Run B Yaw Difference



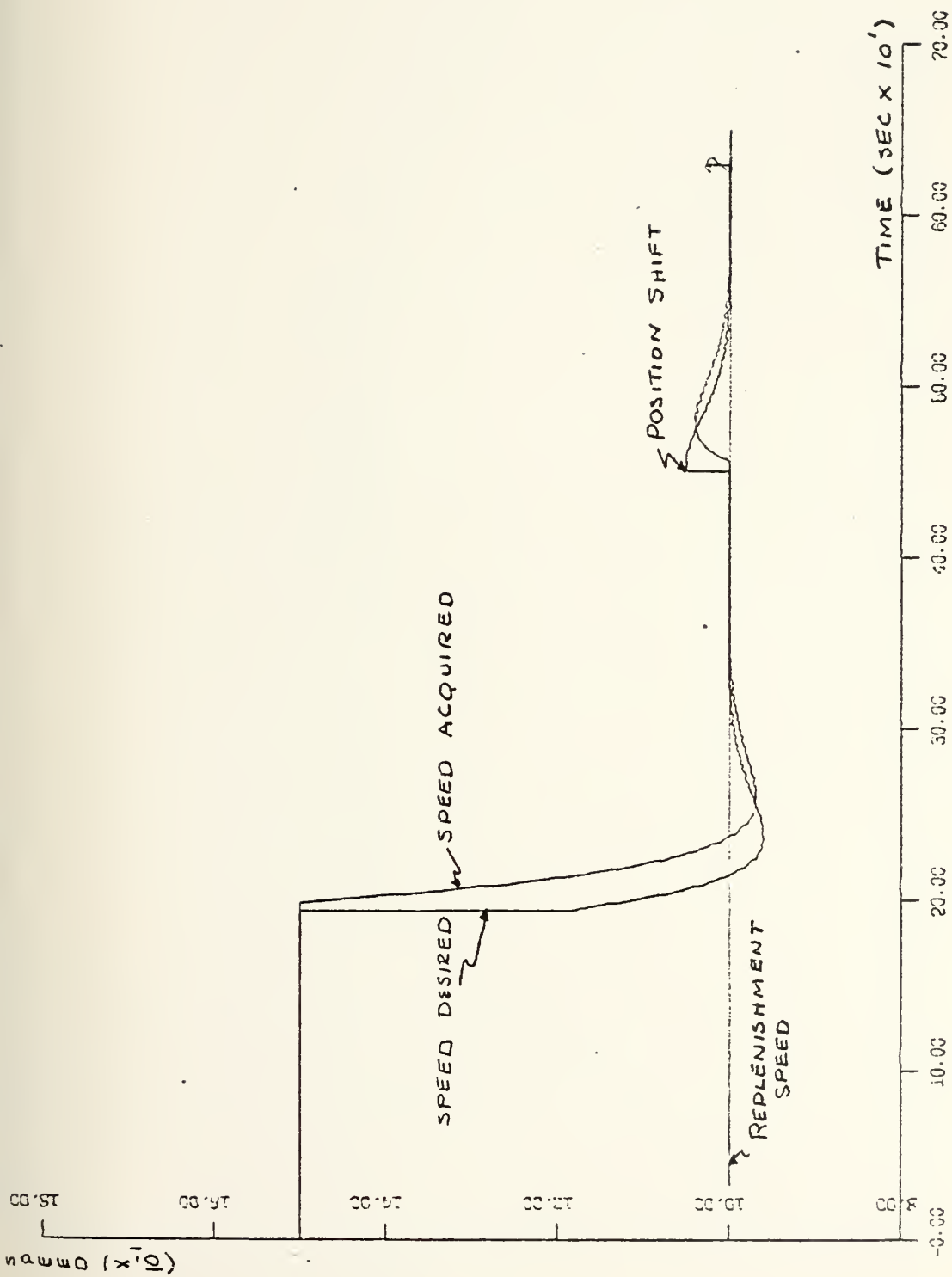


Figure III-88
Approach Phase Run B Speed Response

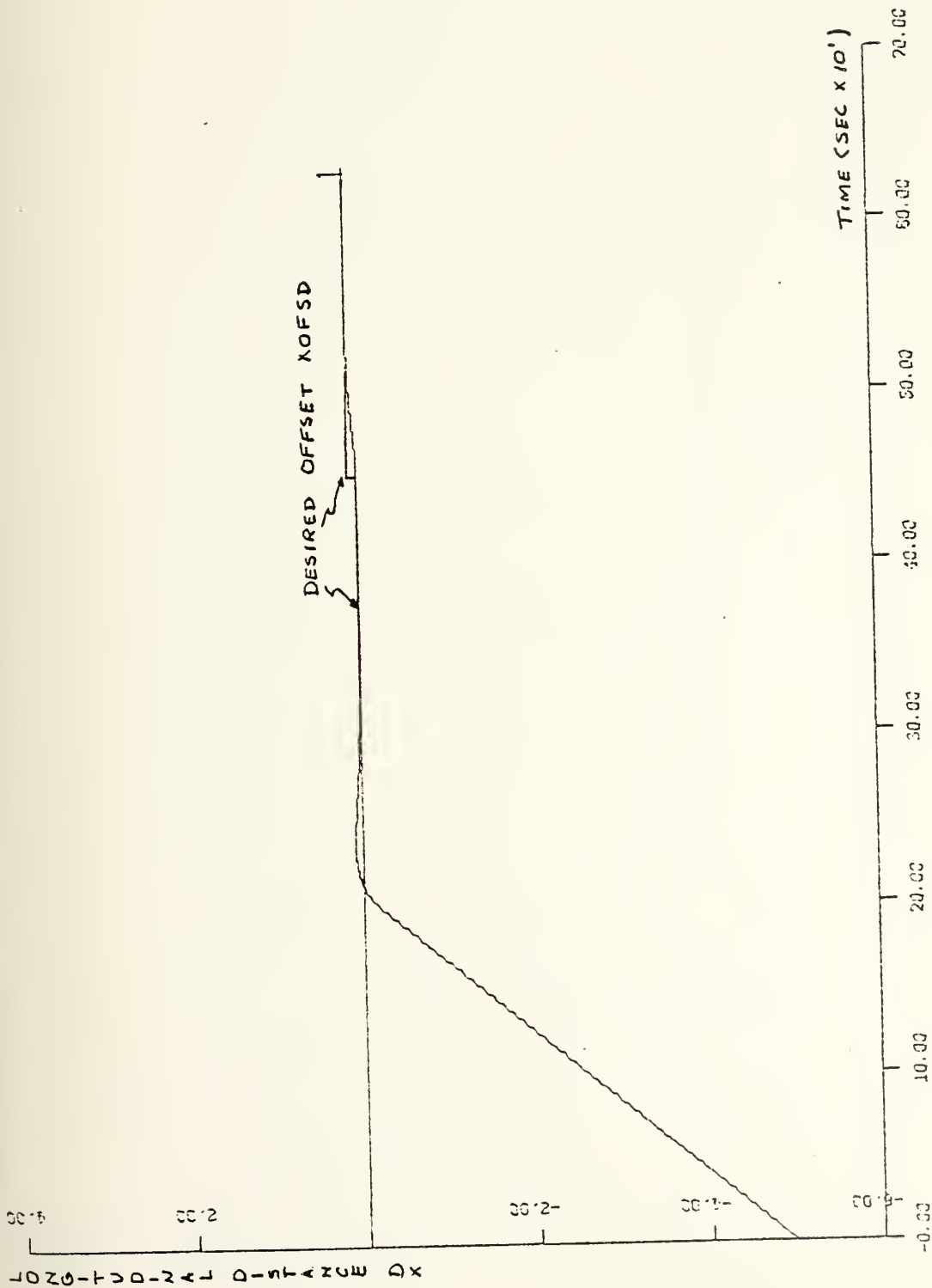


Figure III-89
Approach Phase Run B Longitudinal Position DX

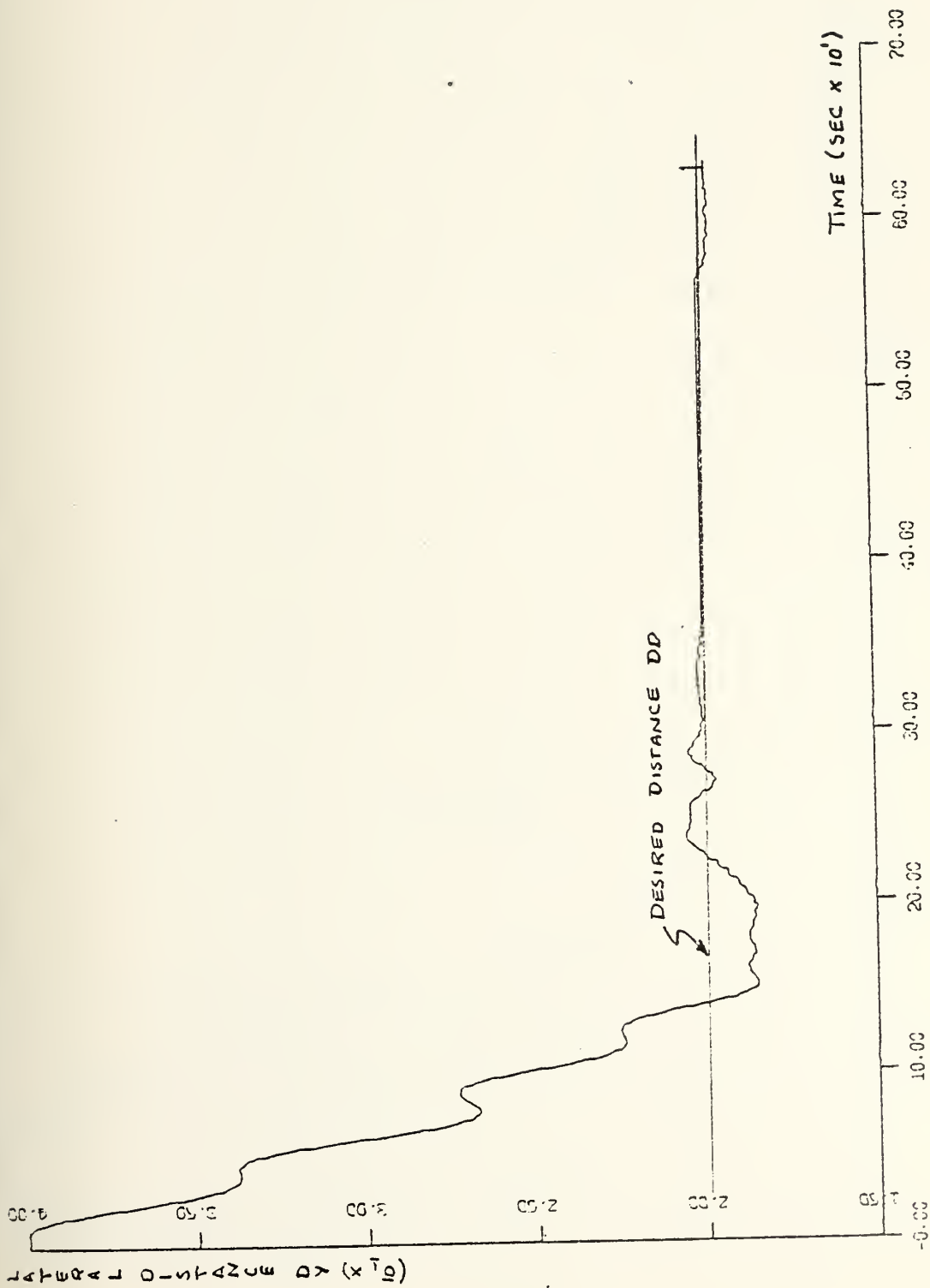


Figure III-90
Approach Phase Run C Lateral Distance DY

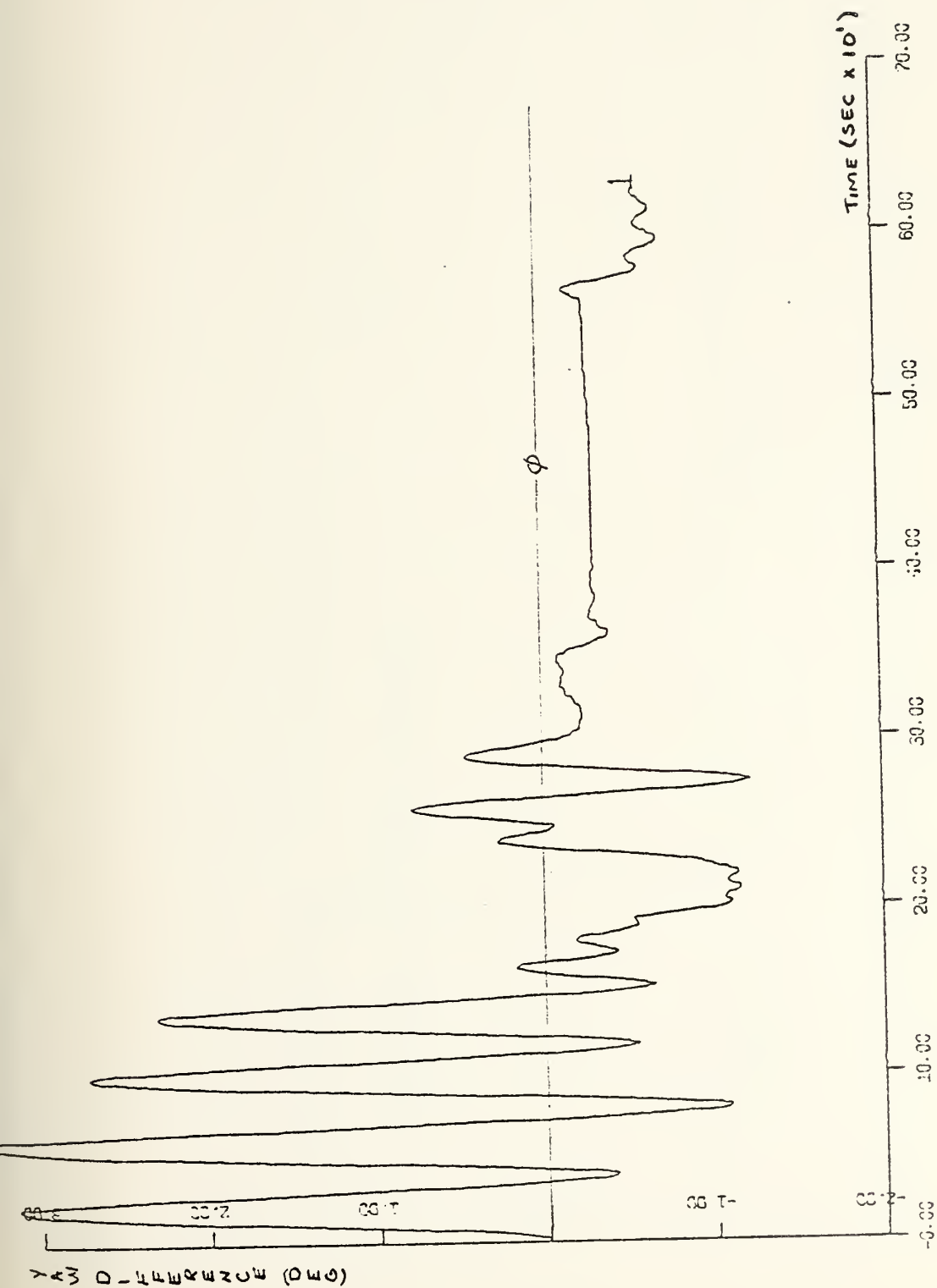


Figure III-91
Approach Phase Run C Yaw Difference

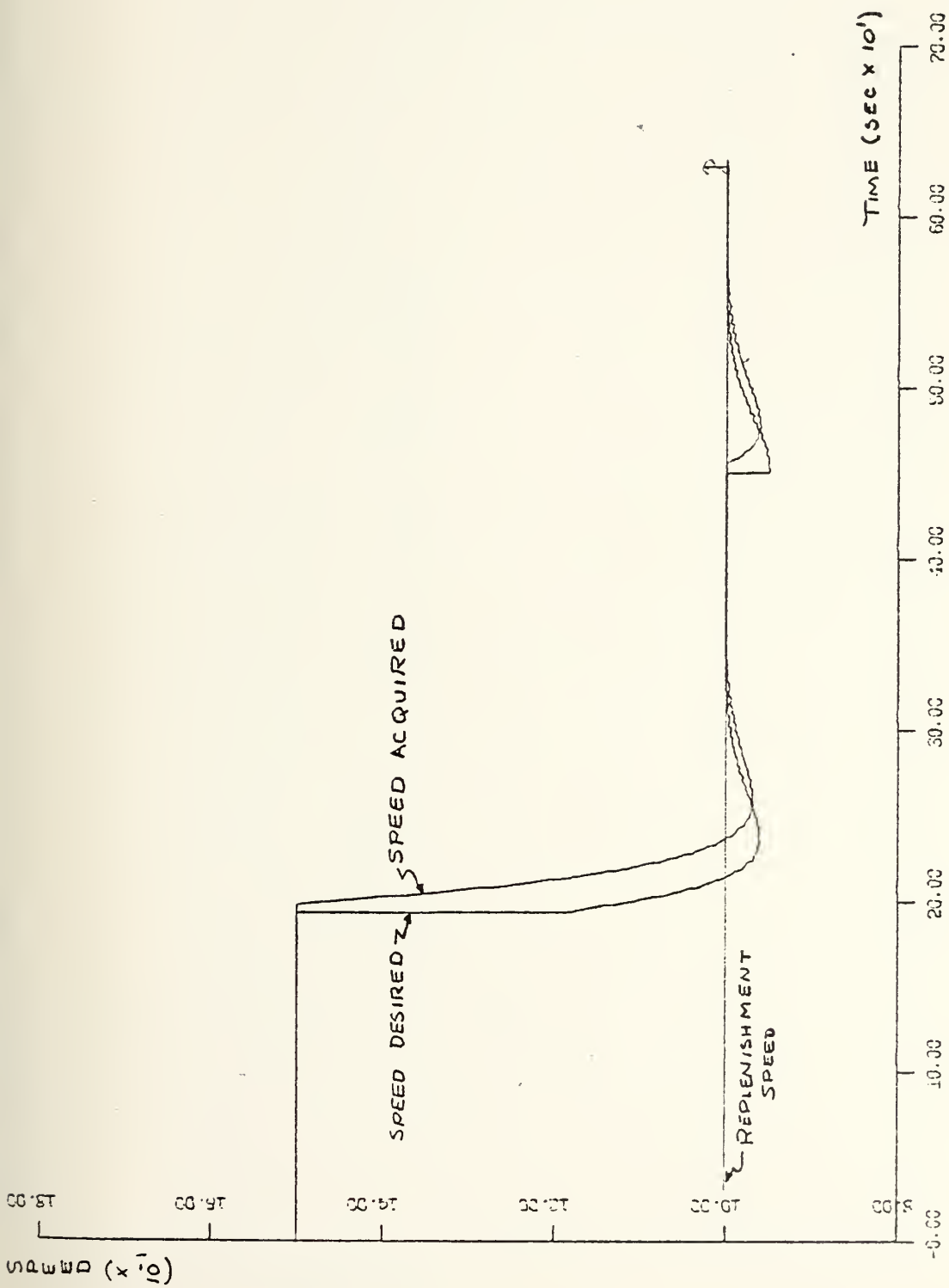


Figure III-92
Approach Phase Run C Speed Response

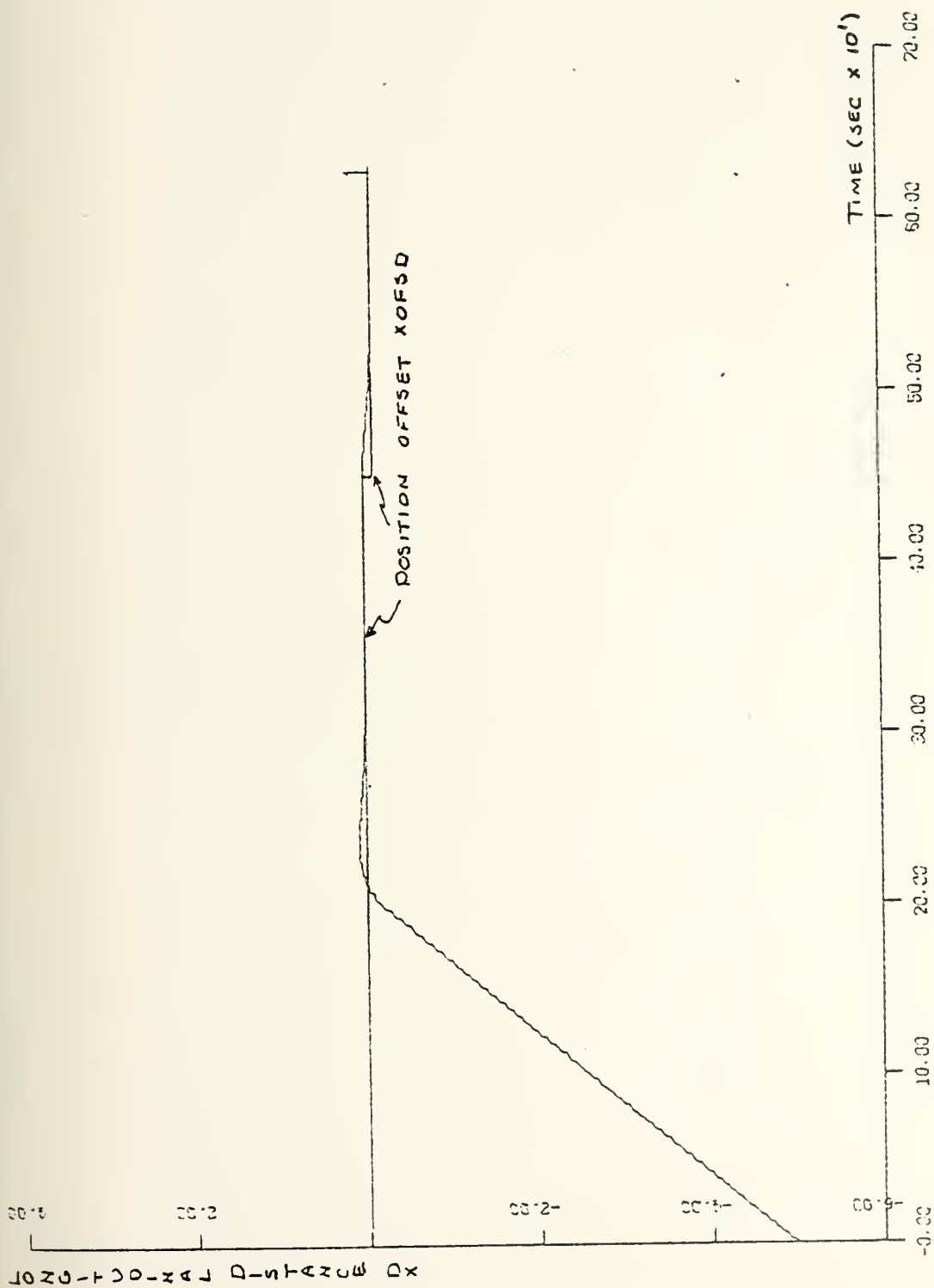


Figure III-93
Approach Phase Run C Longitudinal Position DX

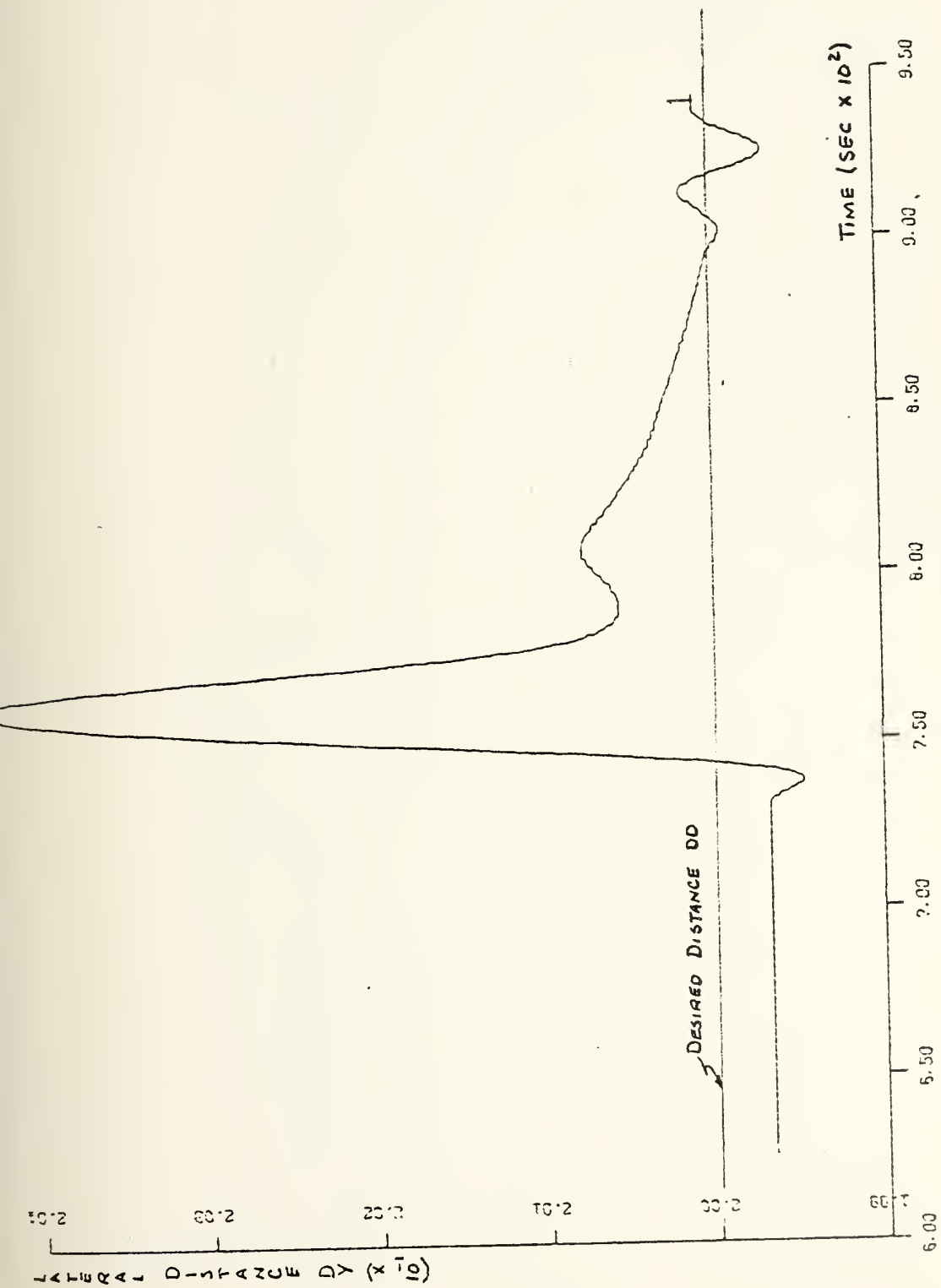
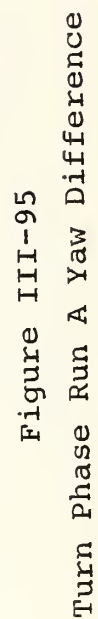


Figure III-94
Turn Phase Run A Lateral Distance DY



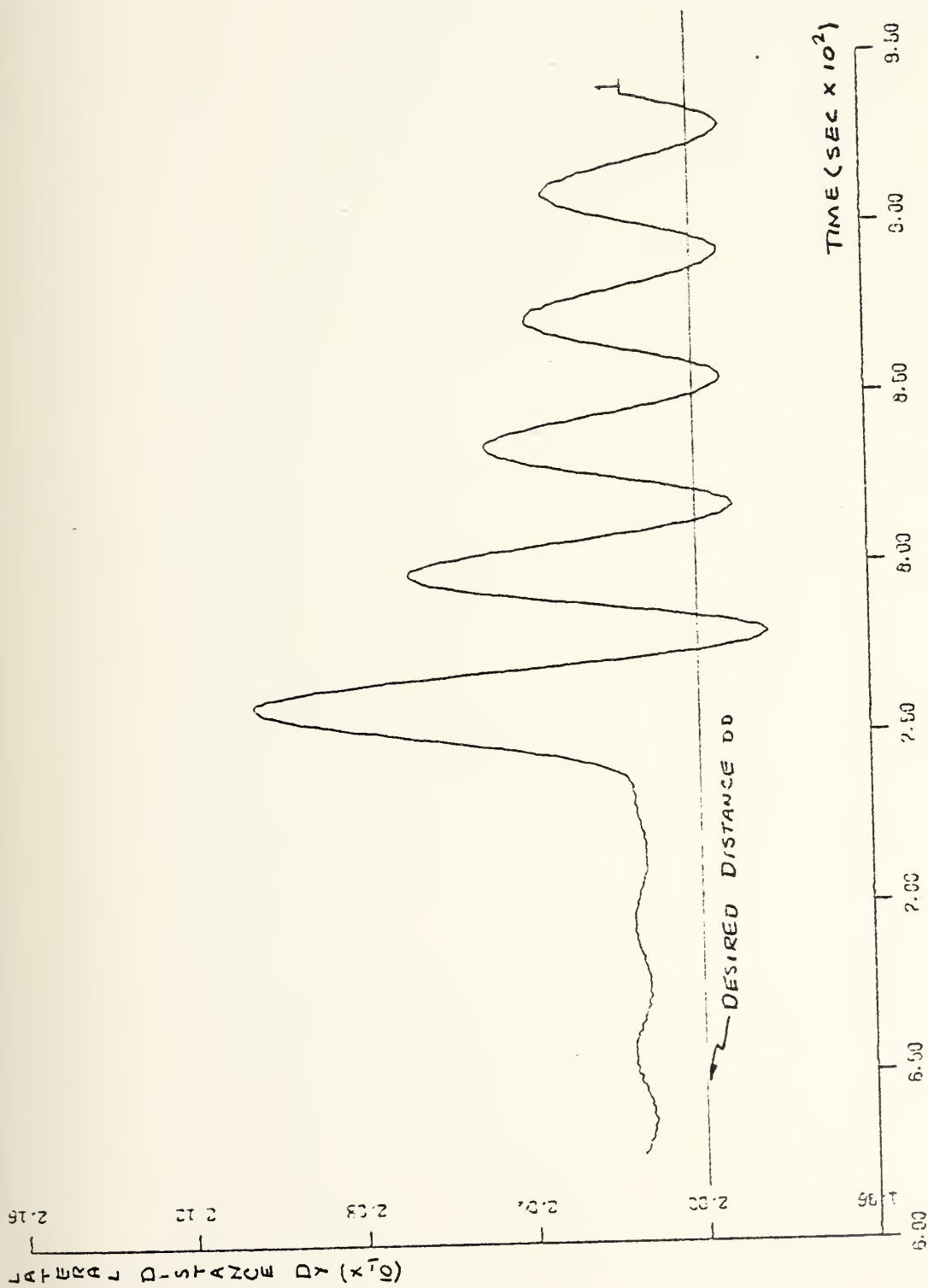


Figure III-96
Turn Phase Run B Lateral Distance DY



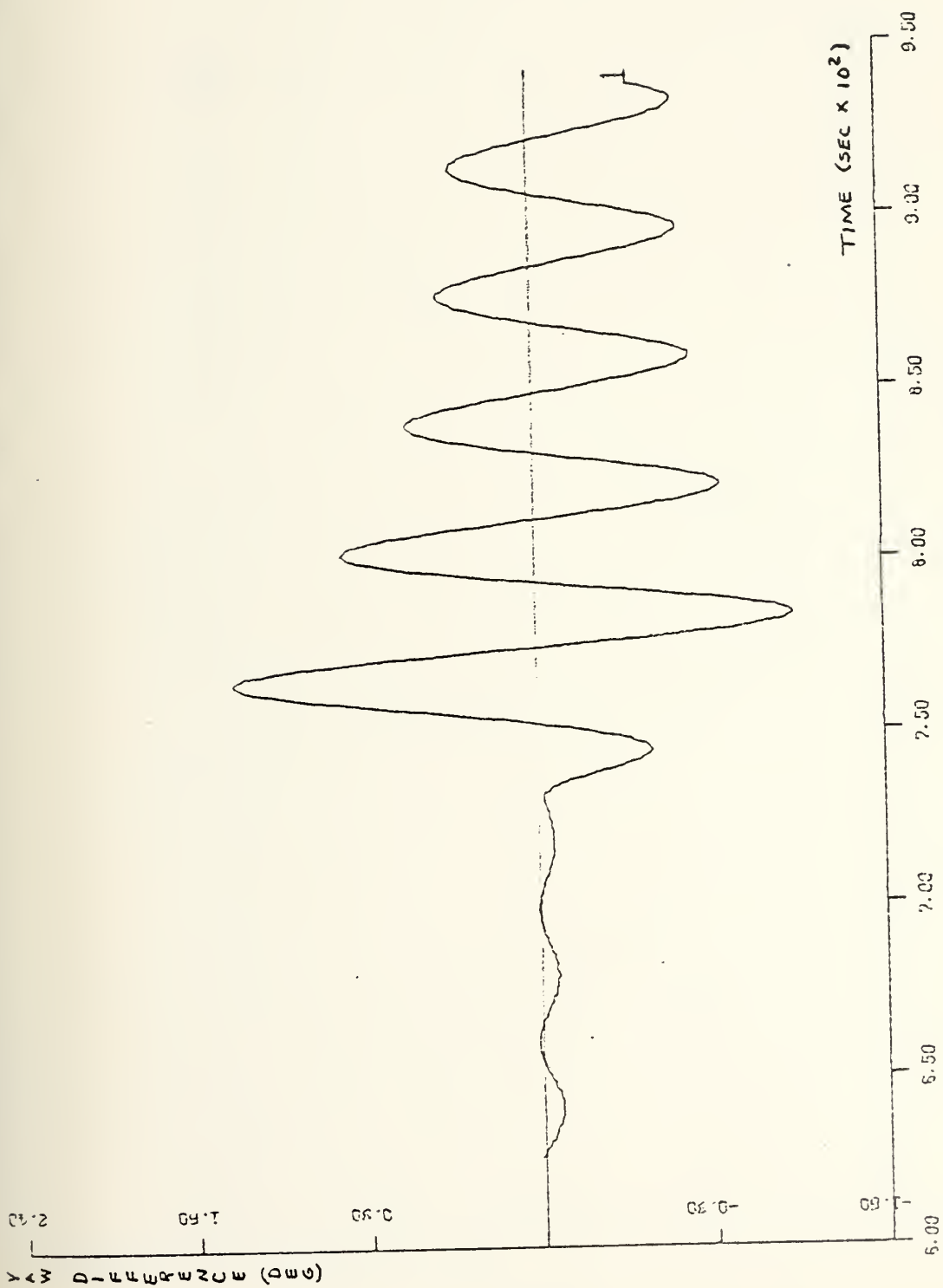


Figure III-97
Turn Phase Run B Yaw Difference

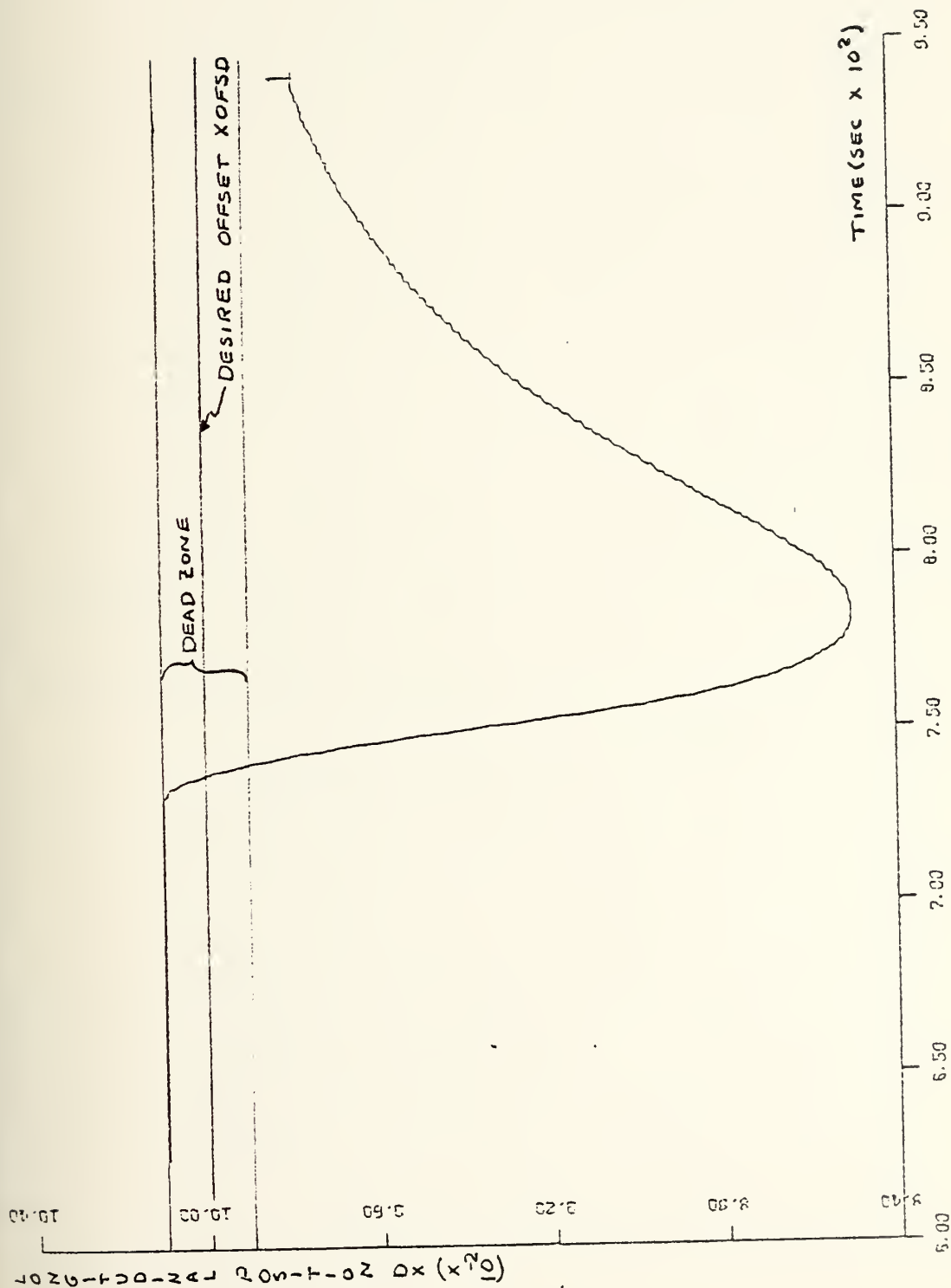


Figure III-98
Turn Phase Run B Longitudinal Position DX

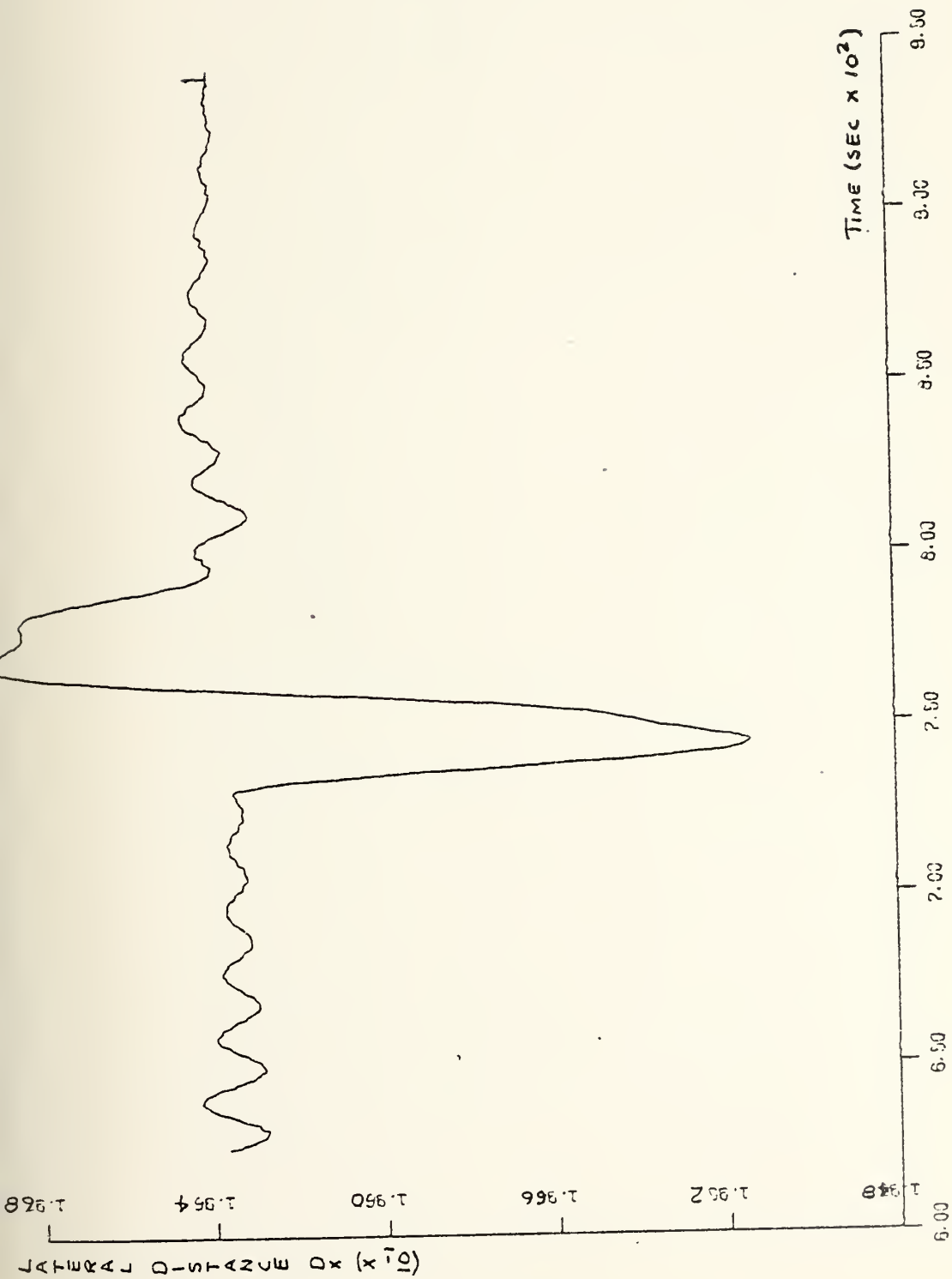


Figure III-99
Turn Phase Run C Lateral Distance DY

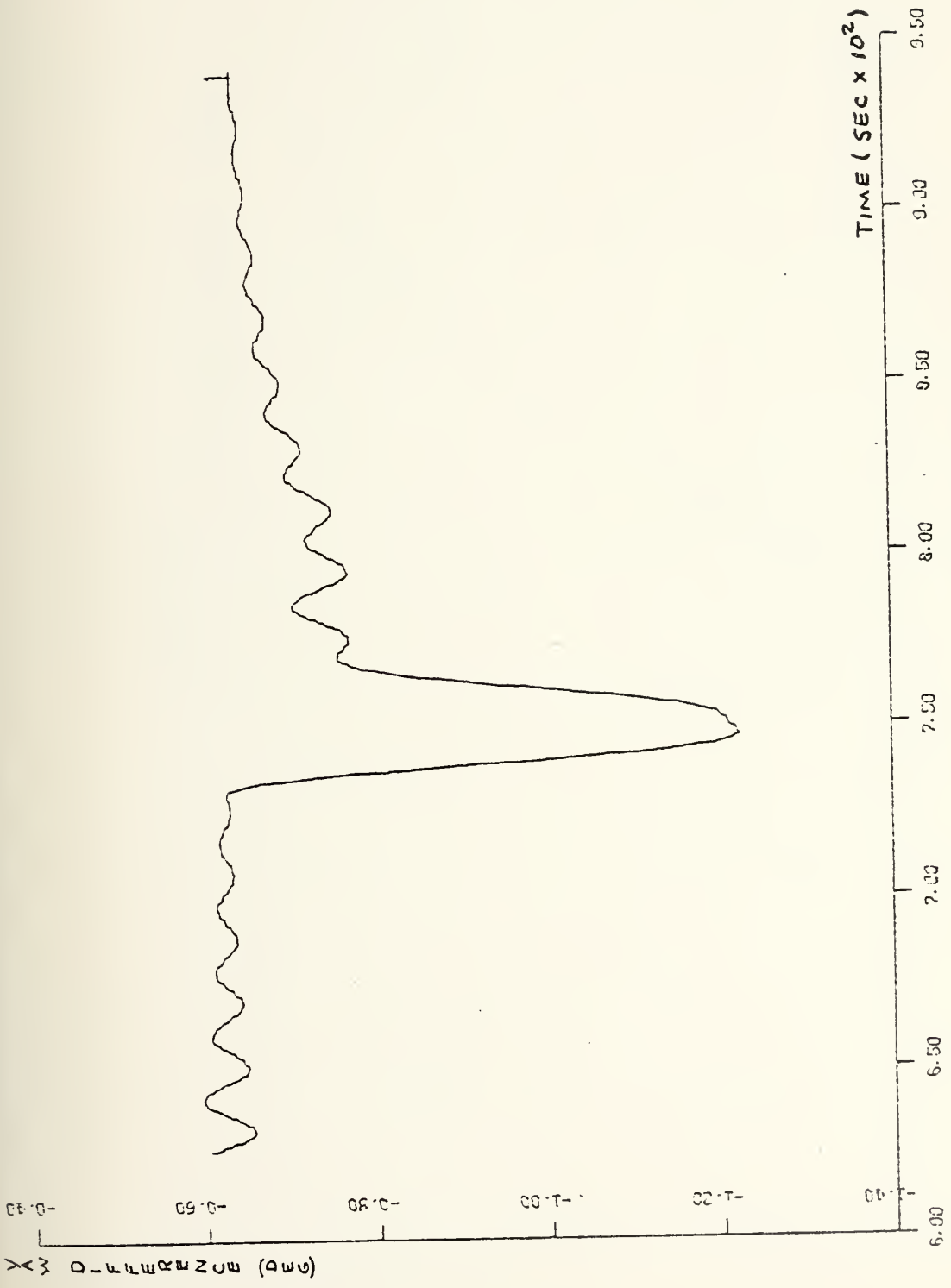


Figure III-100
Turn Phase Run C Yaw Difference

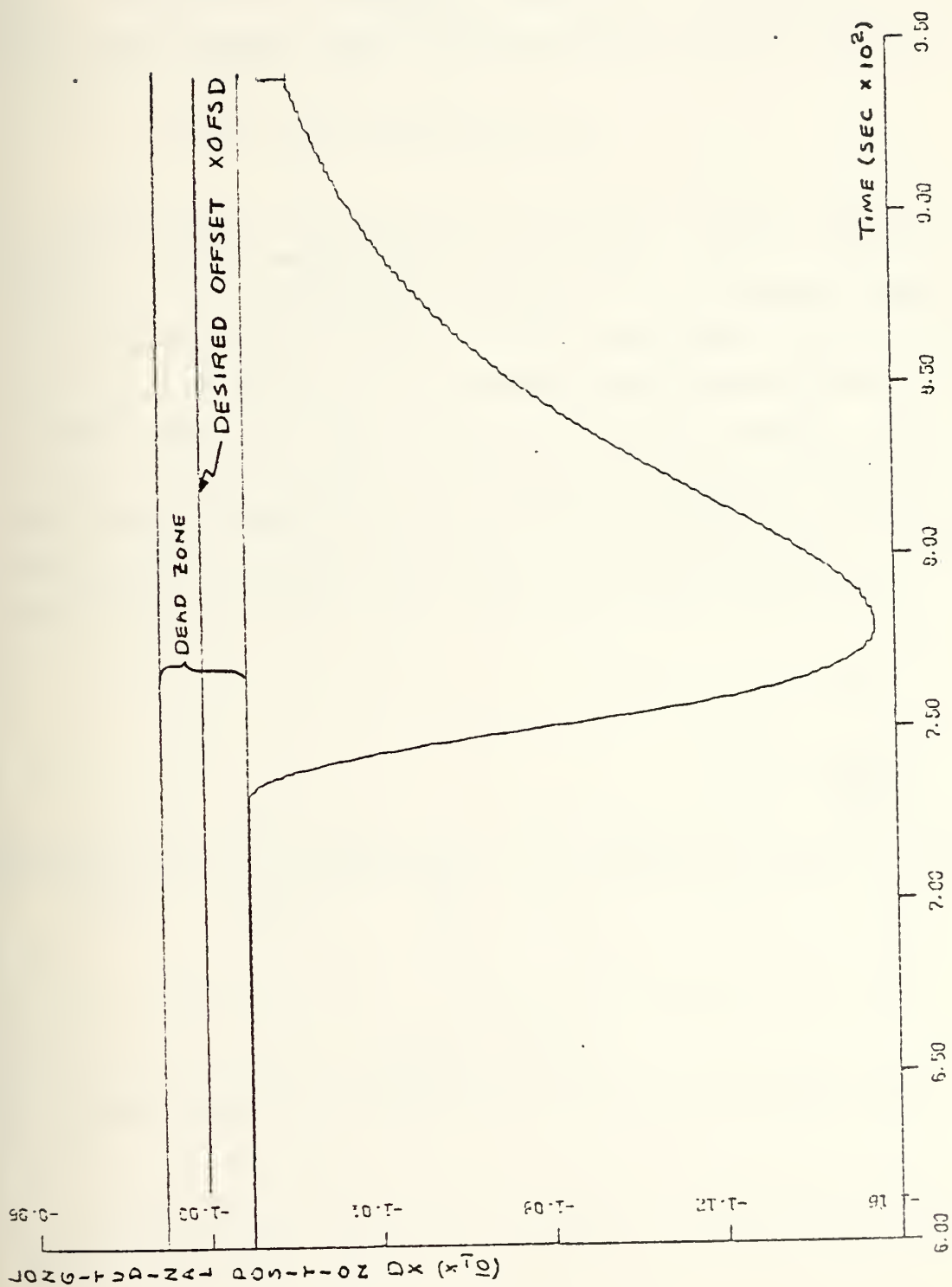


Figure III-101
Turn Phase Run C Longitudinal Position DX

5. Wave Effects on Velocity Control

The final testing procedure involves validation of the speed control system in the presence of waves. This perturbation testing continues that started in section A. of this chapter for heading control. In chapter II the WX force was modeled thru the intermediate force IF32 as:

$$IF32 = KC1 \cdot D2 + NC2 + KC1 \cdot WX$$

By introducing the force in this way, a severe limitation is placed on the magnitude of the force. In the mariner model used, the KC1 coefficient (XDELR) is considered negligible or, at best, only 0.00005. This translates, in the original equations of motion, to a maximum speed perturbation of only 0.0355 kts. for the wave amplitude chosen. The second drawback of this method, with even greater consequences, is that the perturbation is introduced before the control loop. Delay of the wave perturbation is produced making it out of phase with the other wave force (WY) and moment (WN).

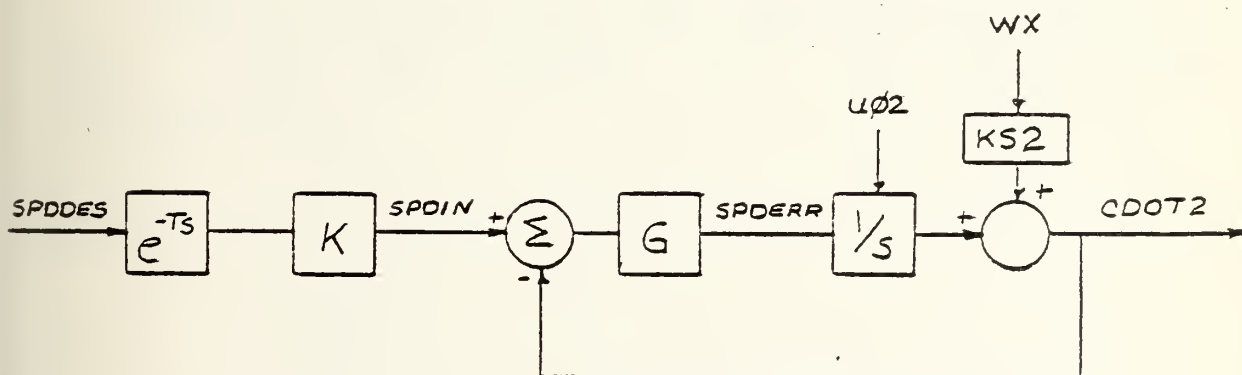


Figure III-102

Block Diagram of Wave Introduction in Speed Loop

In order to bring about uniform introduction of this wave force, its effect is inserted just past the integrator

of the speed control loop as shown in figure III-102. This is coded in the DSI simulation program as:

$$CDOT2 = \text{INTGRL}(U02, SPDERR * LUC) + KS2 * WX$$

A value of -1.0 for KS2 will give a maximum wave perturbation of 0.85275 kts. (a much more realistic perturbation for the high sea state simulated). Figure III-103 portrays the speed desired and speed acquired for the approach phase in the presence of sea state. From this it can be seen that the speed acquired is very dependent upon the sea state present. The control law, however, presents a very stable reference for the speed loop which gives an approach longitudinal position (DX) plot indistinguishable from that of figure III-81. More prominent perturbation results are evident in the turn phase plots of figures III-104 and III-105. The speed response of figure III-104 allows a maximum longitudinal position excursion of 9.5 feet (0.018 normalized position). as compared with 8.286 feet (0.0157 normalized) in calm sea.

These results show that the speed control system is very stable and corrects well for large external perturbations.

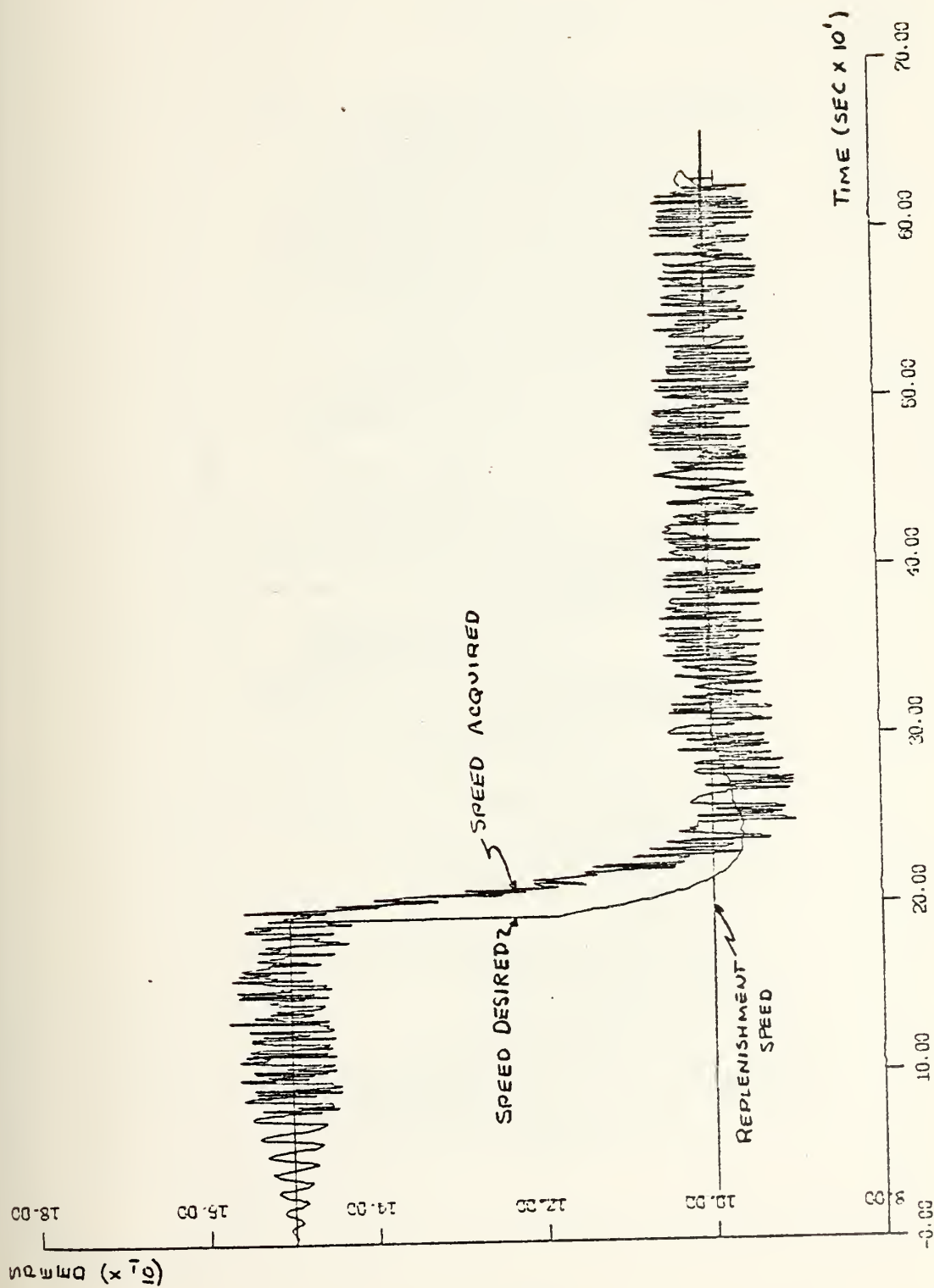


Figure III-103
Approach Phase Speed Response in Waves



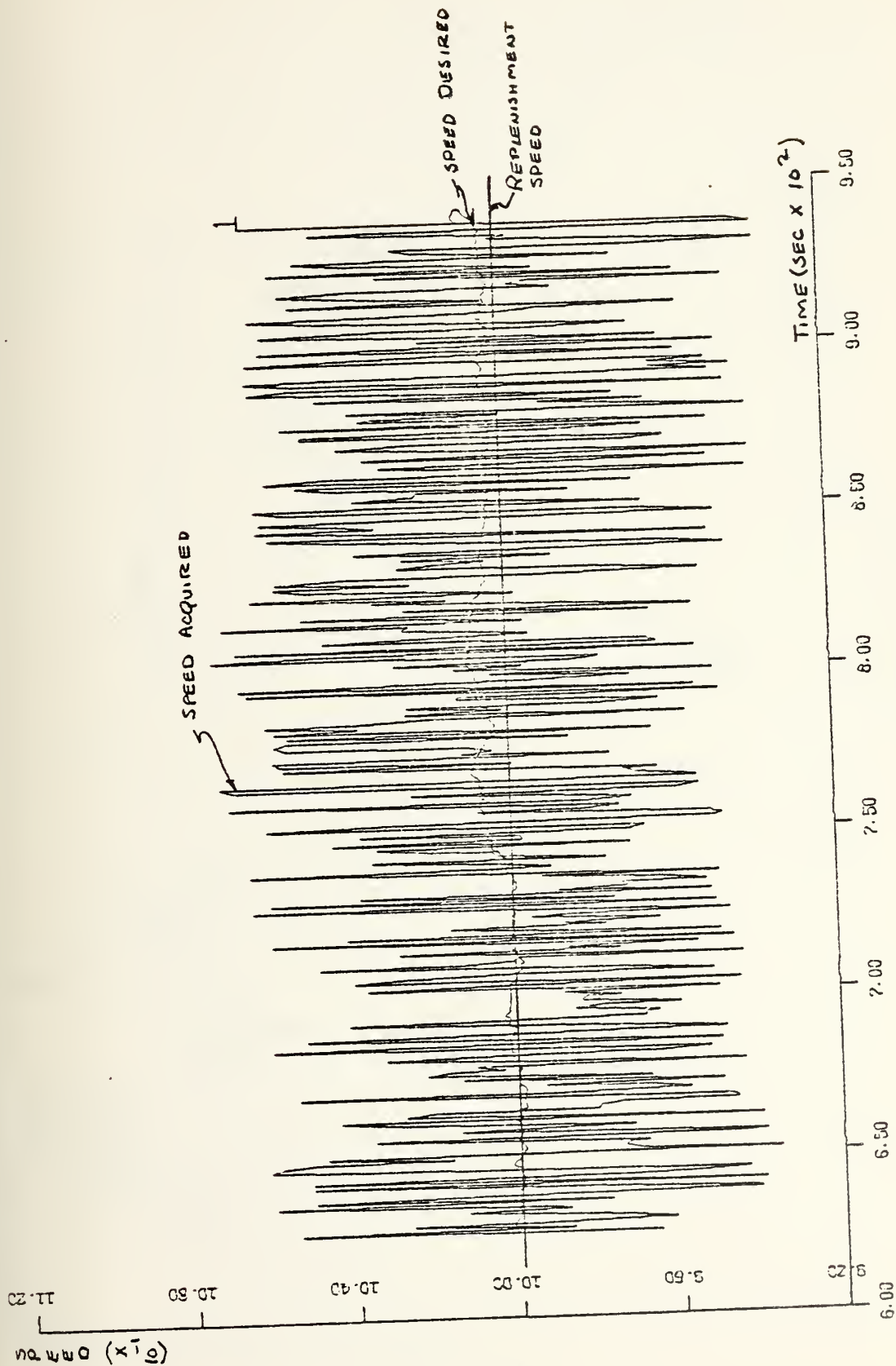


Figure III-104
Turn Phase Speed Response in Waves



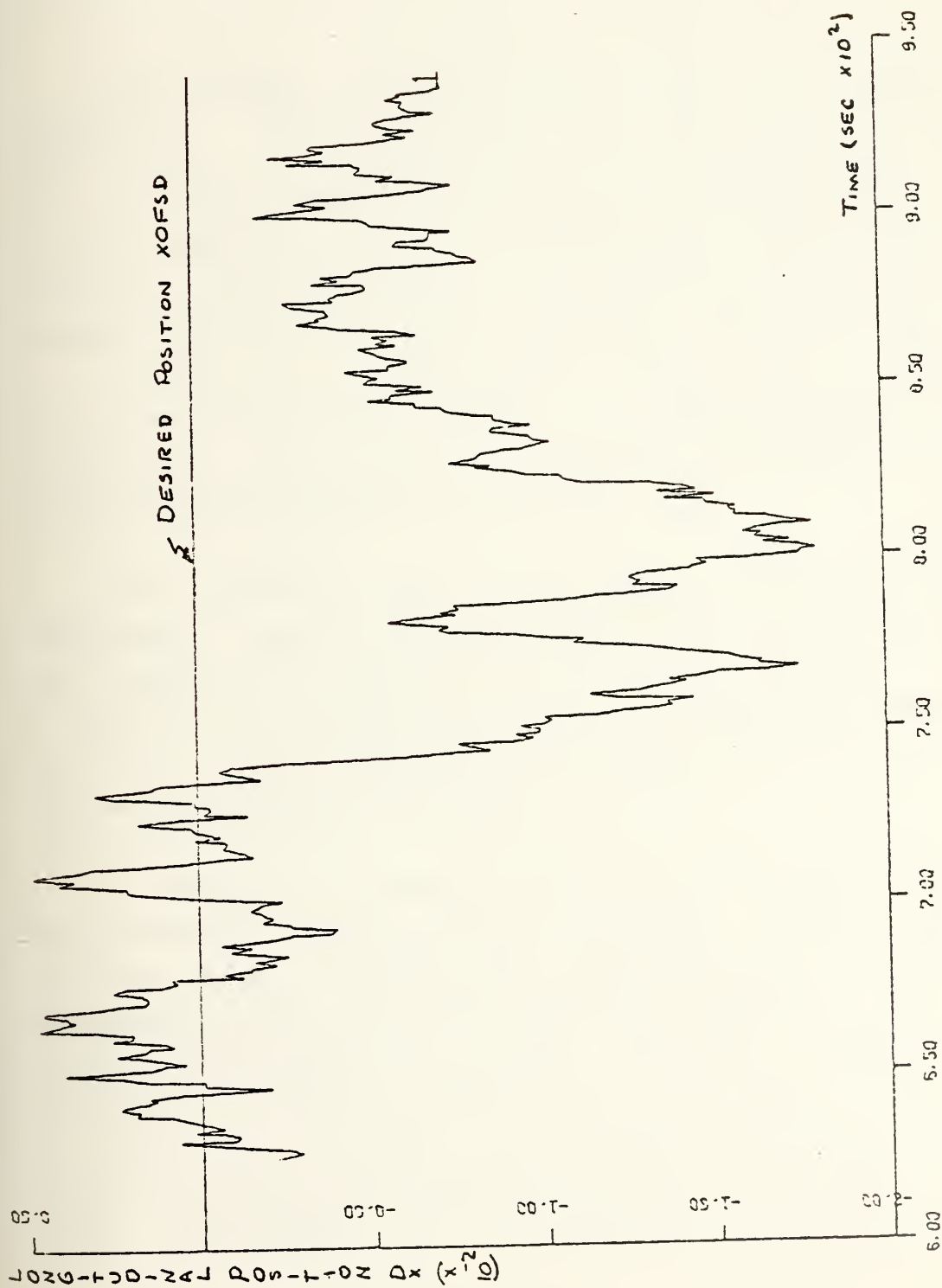


Figure III-105
Turn Phase Longitudinal Position DX in Waves

IV. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The results of this design study have been most gratifying. The basic concepts initially perceived for the RAS control have been realized. The decoupled ship control in the RAS environment is a viable and plausible idea. This thesis contains a workable system for implementation of computer controlled RAS. The achievement of 2.3 foot maximum excursion for lateral distance while both ships are in a turn, and longitudinally offset by 53 feet is a phenomenal achievement. Having this kind of accuracy in RAS operations, can vastly increase the safety of this complicated and dangerous maneuver.

The approach phase of RAS can be a very hair raising experience. Night replenishment and sea state complicate the "seaman's eye" method now employed in the fleet. Having a system that automatically handles the approach regardless of the adverse conditions can, again, do nothing more than increase the safety of the RAS maneuver.

Schemes for computer control of nonlinear systems and the purposeful introduction of nonlinear control laws are becoming more practical with the technological advances in micro processors. The ever increasing number of U.S. Navy ships with computer systems installed, makes digital computer ship control realizable in the present time frame. A good micro computer or an existing installed computer (such as one used for the NTDS system) can be used in this vein. Procurement of the hardware required for this RAS

system can be dissipated over time periods contingent on the funding available. The supply ship requires only two reflectors for the range and bearing devices stationed on the receiving ship. All ships can be outfitted with such reflectors at a minimal cost, while the bulk of the hardware can be introduced to the ships at regularly scheduled yard periods.

In the initial conception of this thesis, a section was planned for open ocean maneuvering. After some research on this facet of ship control, it was determined that work in this area has already been documented[²⁴][²⁵]. The existence of NTDS outputs for station attainment and single ship control systems, made design in this area redundant.

The concept of integrated centralized ship control has been in the background for over a decade[²⁶]. Although given a low priority due to funding considerations, its implementation seems to be just around the corner[²⁷]. However, a review of ref. 27 indicates a lack of RAS capability. Whether this is an oversight in the article or neglected in the design criteria is unknown. If it has been neglected in the design, a very real problem has been overlooked. The recent incidences of ship collisions while conducting RAS [²⁸] emphasizes the need for inclusion of this very dangerous maneuver in the "Integrated Bridge System." Lack of technology can no longer be used as an excuse. This thesis and other research reports [²⁹] have advanced the implementation feasibility to a level that cannot be ignored. With these projects finalized in practical terms, their incorporation into fleet use is the next imperative step.

A major effort in this area must be made. The ever increasing complexities of today's naval ships and the loads being placed on the officers and men are such that computer

control must be used; and used now! We cannot afford the luxury of time to prove these systems^① worth, but must make concerted efforts to get them implemented before the lives of 300+ men are lost.

Whenever a complicated system such as a ship in the RAS situation is encountered, many facets have to be concurrently analyzed. This fact has caused inclusion of many diagrams in this thesis to illustrate the total picture. Each run, with a different condition, requires many plots to analyze the differences in the responses and the causes of the differences. The computer programs shown do not reflect the actual run times in the JCL shown. As many as twenty plots were output in these programs in the times listed. Analysis of the actual computation times show that the algorithms run considerably under the time required for real time operation. The sampled data rate used in the simulations was 0.11 seconds. This is well within the realizable data rates available in even the slowest of today's computers and microprocessors. The thrust of this consideration is that there are no problems envisioned in converting RAS simulation to real world RAS control.

E. RECOMMENDATIONS

In the heading control design section of this thesis, the need for a completely adaptive gain scheme was cited. Again in the velocity control section, when a longitudinal offset was introduced, this need became even more evident. The first and most important recommendation for further study is the development of just such an adaptive gain scheme.

The linear equations of motion should be replaced with nonlinear equations to validate the control designs advanced

in this thesis. Along with this, the hydrodynamic coefficients for the Navy's modern ships are required to be able to design these control systems for today's vessels.

It is further recommended that a concerted effort be made to obtain data on the interactive forces and moments between ships of dissimilar types and sizes. These forces and moments must also be available for sea state conditions. In fact, the whole area on sea state effects on the various ship types in the RAS situation and in open ocean maneuvering needs attention. Not enough data was available for this researcher to be able to pinpoint sea state effects on ship hulls. Since replenishment at sea is rarely conducted in the sterile condition of calm sea, these considerations are of utmost importance to allow testing of any control system in the simulation stage of development.

The intent here is not to imply that the control systems portrayed in this thesis are the best for the RAS scenario, but that the procedures used can be applied to any control scheme desired and benchmarked to the ones contained here. As previously mentioned, much meaningful research and design must be accomplished to allow system reliability and, more important, system acceptability by the officers and men who will ultimately trust their lives to it. This is a task that must not be taken lightly.

APPENDIX A

Due to the lengthy nature of the computer programs presented in this thesis, many functions and subroutines were developed to simplify their presentation. This appendix lists these functions and subroutines in alphabetical order. The computer programs reference this appendix and indicate the placement of the required functions and subroutines.

A brief description for each listing is given to aid the reader in determining their purpose and use. The following is a listing of the functions and subroutines contained in this appendix in the order presented:

SUBROUTINE BOXPLX
FUNCTION DEGRAD
FUNCTION DELAY
FUNCTION FE - RUN A (FEA)
FUNCTION FE - RUN E (FEB)
FUNCTION FE - RUN C (FEC)
SUBROUTINE HDGRAS
FUNCTION KE
MAIN PROGRAM FOR FUNCTION MINIMIZATIONS (MINIEXPX)
SUBROUTINE RBMEAS
FUNCTION RKLDEQ
SUBROUTINE SLOPES
FUNCTION SPINIT
FUNCTION SPDCTR
FUNCTION SEDOFC
FUNCTION SPDREC
FUNCTION SWCL
SUBROUTINE SWITCH

SUBROUTINE SWTCHF
SUBROUTINE TRANS
FUNCTION XLIMIT

SUBROUTINE EOXPX

This subroutine was used for all optimization runs in heading control and speed control. It was programmed locally and is part of the IBM 360 SSP library at the Naval Postgraduate School. A full explanation and description is shown in the first few pages of the subroutine listing.

SUBROUTINE BOXPLX

.....
SUBROUTINE BOXPLX (CATEGORY H0)
PURPOSE
BOXPLX IS A SUBROUTINE USED TO SOLVE THE PROBLEM OF LOCATING
A MINIMUM (OR MAXIMUM) OF AN ARBITRARY OBJECTIVE FUNCTION
SUBJECT TO ARBITRARY EXPLICIT AND/OR IMPLICIT CONSTRAINTS BY
THE COMPLEX METHOD OF M.J. BOX. EXPLICIT CONSTRAINTS ARE
DEFINED AS UPPER AND LOWER BOUNDS ON THE INDEPENDENT VARIABLES.
IMPLICIT CONSTRAINTS MAY BE ARBITRARY FUNCTIONS OF THE VAR-
IABLES. TWO FUNCTION SUBPROGRAMS TO EVALUATE THE OBJECTIVE
FUNCTION AND IMPLICIT CONSTRAINTS, RESPECTIVELY, MUST BE
SUPPLIED BY THE USER (SEE EXAMPLE PROGRAMMING). BOXPLX ALSO HAS
THE OPTION TO PERFORM INTEGER PROGRAMMING, WHERE THE VALUES
OF THE INDEPENDENT VARIABLES ARE RESTRICTED TO INTEGERS.
USAGE
CALL BOXPLX (NV,NAV,NPR,NTA,R,XS,IP,XU,XL,YMN,IER)
DESCRIPTION OF PARAMETERS
NV AN INTEGER INPUT DEFINING THE NUMBER OF INDEPENDENT
VARIABLES OF THE OBJECTIVE FUNCTION TO BE MINIMIZED.
NOTE: MAXIMUM NV + NAV IS PRESENTLY 50. MAXIMUM NV IS
25. IF THESE LIMITS MUST BE EXCEEDED, PUNCH A SOURCE
DECK IN THE USUAL MANNER, AND CHANGE THE DIMENSION
STATEMENTS.
NAV AN INTEGER INPUT DEFINING THE NUMBER OF AUXILIARY VAR-
IABLES THE USER WISHES TO DEFINE FOR HIS OWN CONVENIENCE.
TYPICALLY HE MAY WISH TO DEFINE THE VALUE OF EACH IMPLICIT
CONSTRAINT FUNCTION AS AN AUXILIARY VARIABLE. IF THIS
IS DONE, THE OPTIONAL OUTPUT FEATURE OF BOXPLX CAN BE
USED TO OBSERVE THE VALUES OF THOSE CONSTRAINTS AS THE
SOLUTION PROGRESSES. AUXILIARY VARIABLES, IF USED,
SHOULD BE EVALUATED IN FUNCTION KE (DEFINED BELOW).
NAV MAY BE ZERO.
NPR INPUT INTEGER CONTROLLING THE FREQUENCY OF OUTPUT DESIRED
.....
BXPX0010
BXPX0020
BXPX0030
BXPX0040
BXPX0050
BXPX0060
BXPX0070
BXPX0080
BXPX0090
BXPX0100
BXPX0110
BXPX0120
BXPX0130
BXPX0140
BXPX0150
BXPX0160
BXPX0170
BXPX0180
BXPX0190
BXPX0200
BXPX0210
BXPX0220
BXPX0230
BXPX0240
BXPX0250
BXPX0260
BXPX0270
BXPX0280
BXPX0290
BXPX0300
BXPX0310
BXPX0320
BXPX0330
BXPX0340
BXPX0350
BXPX0360
BXPX0370
BXPX0380
BXPX0390
BXPX0400
BXPX0410
BXPX0420
BXPX0430

	FOR DIAGNOSTIC PURPOSES. IF NPR .LE. 0, NO OUTPUT WILL BE PRODUCED BY BOXPLX. OTHERWISE, THE CURRENT COMPLEX OF K= 2*NV VERTICES AND THEIR CENTROID WILL BE OUTPUT AFTER EACH NPR PERMISSIBLE TRIALS. THE NUMBER OF TOTAL TRIALS, NUMBER OF FEASIBLE TRIALS, NUMBER OF FUNCTION EVALUATIONS, AND NUMBER OF IMPLICIT CONSTRAINT EVALUATIONS ARE IN- CLUDED IN THE OUTPUT. ADDITIONALLY, (WHEN NPR .GT. 0) THE SAME INFORMATION WILL BE OUTPUT:	BEPX0440 BXPX0450 BXPX0460 BXPX0470 BXPX0480 BXPX0490 BXPX0500 BXPX0510 BXPX0520 BXPX0530 BXPX0540 BXPX0550 BXPX0560 BXPX0570 BXPX0580 BXPX0590 BXPX0600 BXPX0610 BXPX0620 BXPX0630 BXPX0640 BXPX0650 BXPX0660 BXPX0670 BXPX0680 BXPX0690 BXPX0700 BXPX0710 BXPX0720 BXPX0730 BXPX0740 BXPX0750 BXPX0760 BXPX0770 BXPX0780 BXPX0790 BXPX0800 BXPX0810 BXPX0820 BXPX0830 BXPX0840 BXPX0850 BXPX0860 BXPX0870 BXPX0880 BXPX0890 BXPX0900
	1) IF THE INITIAL POINT IS NOT FEASIBLE, GENERATED, 2) AFTER THE FIRST COMPLETE COMPLEX IS GENERATED, 3) IF A FEASIBLE VERTEX CANNOT BE FOUND AT SOME TRIAL, 4) IF THE OBJECTIVE VALUE OF A VERTEX CANNOT BE MADE NO-LONGER-WORST. 5) IF THE LIMIT ON TRIALS (NTA) IS REACHED AND, 6) WHEN THE OBJECTIVE FUNCTION HAS BEEN UNCHANGED FOR 2*NV TRIALS, INDICATING A LOCAL MINIMUM HAS BEEN FOUND.	
	IF THE USER WISHES TO TRACE THE PROGRESS OF A SOLUTION, A CHOICE OF NPR = 25, 50 OR 100 IS RECOMMENDED.	
NTA	INTEGER INPUT OF LIMIT ON THE NUMBER OF TRIALS ALLOWED IN THE CALCULATION. IF THE USER INPUTS NTA .LE. 0, A DEFAULT VALUE OF 2000 IS USED. WHEN THIS LIMIT IS REACHED CONTROL RETURNS TO THE CALLING PROGRAM WITH THE BEST ATTAINED OBJECTIVE FUNCTION VALUE IN YMN, AND THE BEST ATTAINED SOLUTION POINT IN XS.	
R	A REAL NUMBER INPUT TO DEFINE THE FIRST RANDOM NUMBER USED IN DEVELOPING THE INITIAL COMPLEX OF 2*NV VERTICES. (0. .GT. R .LT. 1.) IF R IS NOT WITHIN THESE BOUNDS, IT WILL BE REPLACED BY 1./3. .	
XS	INPUT REAL ARRAY DIMENSIONED AT LEAST NV+NAV. THE FIRST NV MUST CONTAIN A FEASIBLE ORIGIN FOR STARTING THE CAL- CULATION. THE LAST NAV NEED NOT BE INITIALIZED. UPON RETURN FROM BOXPLX, THE FIRST NV ELEMENTS OF THE ARRAY CONTAIN THE COORDINATES OF THE MINIMUM OBJECTIVE FUNCTION, AND THE REMAINING NAV (NAV-GE. 0) CONTAIN THE VALUES OF THE CORRESPONDING AUXILIARY VARIABLES.	
IP	INTEGER INPUT FOR OPTIONAL INTEGER PROGRAMMING. IF IP=1, THE VALUES OF THE INDEPENDENT VARIABLES WILL BE REPLACED WITH INTEGER VALUES (STILL STORED AS REAL*4).	
XU	A REAL ARRAY DIMENSIONED AT LEAST NV INPUTTING THE UPPER	BXPX0910

CC







BX PX 2360
BX PX 2370
BX PX 2380
BX PX 2390
BX PX 2400
BX PX 2410
BX PX 2420
BX PX 2430
BX PX 2440
BX PX 2450
BX PX 2460
BX PX 2470
BX PX 2480
BX PX 2490
BX PX 2500
BX PX 2510
BX PX 2520
BX PX 2530
BX PX 2540
BX PX 2550
BX PX 2560
BX PX 2570
BX PX 2580
BX PX 2590
BX PX 2600
BX PX 2610
BX PX 2620
BX PX 2630
BX PX 2640
BX PX 2650
BX PX 2660
BX PX 2670
BX PX 2680
BX PX 2690
BX PX 2700
BX PX 2710
BX PX 2720
BX PX 2730
BX PX 2740
BX PX 2750
BX PX 2760
BX PX 2770
BX PX 2780
BX PX 2790
BX PX 2800
BX PX 2810
BX PX 2820
BX PX 2830

CLARING "IP" TO BE AN ARRAY OF NV CONTROL VARIABLES WHERE IP (I)=1 WOULD INDICATE THAT THE I-TH VARIABLE IS TO BE CONFINED TO INTEGER VALUES. EACH STATEMENT OF THE FORM IF (IP.EQ.I), ETC. WOULD THEN NEED TO BE ALTERED TO IF (IP(I).EQ.I), ETC., WHERE THE SUBSCRIPT IS APPROPRIATELY CHOSEN. NORMALLY, XU AND XL VALUES ARE ALTERED TO BE AN EPSILON WITHIN ACTUAL VALUES DECLARED BY THE USER. THIS ADJUSTMENT IS NOT MADE WHEN IP=1.

NOTE: NO NON-LINEAR PROGRAMMING ALGORITHM CAN GUARANTEE THAT THE ANSWER FOUND IS THE GLOBAL MINIMUM, RATHER THAN JUST A LOCAL MINIMUM. HOWEVER, ACCORDING TO REF. 2, THE COMPLEX METHOD HAS AN ADVANTAGE IN THAT IT TENDS TO FIND THE GLOBAL MINIMUM MORE FREQUENTLY THAN MANY OTHER NON-LINEAR PROGRAMMING ALGORITHMS.

IT SHOULD BE NOTED THAT THE AUXILIARY VARIABLE FEATURE CAN ALSO BE USED TO DEAL WITH PROBLEMS CONTAINING EQUALITY CONSTRAINTS. ANY EQUALITY CONSTRAINT IMPLIES THAT A GIVEN VARIABLE IS NOT TRULY INDEPENDENT. THEREFORE, IN GENERAL, ONE VARIABLE INVOLVED IN AN EQUALITY CONSTRAINT CAN BE RENUMBERED FROM THE SET OF NV INDEPENDENT VARIABLES AND ADDED TO THE SET OF NAV AUXILIARY VARIABLES. THIS USUALLY INVOLVES RENUMBERING THE INDEPENDENT VARIABLES OF THE GIVEN PROBLEM.

SUBROUTINES AND FUNCTIONS REQUIRED

SUBROUTINE 'BOUN' AND FUNCTION 'FBV' ARE INTEGRAL PARTS OF THE BOXPLX PACKAGE.

TWO FUNCTIONS MUST BE SUPPLIED BY THE USER. THE FIRST, KE(X), IS USED TO EVALUATE THE IMPLICIT CONSTRAINTS. SET KE=0 AT THE BEGINNING OF THE FUNCTION, THEN EVALUATE THE IMPLICIT CONSTRAINTS. IN THE EXAMPLE ABOVE, THE FIRST CONSTRAINT, X(3), MUST BE WITHIN THE RANGE (0..LE. 6.). THE SECOND CONSTRAINT X(4), MUST BE GE. 0. IF EITHER CONSTRAINT IS NOT WITHIN THESE BOUNDS, CONTROL IS TRANSFERRED TO STATEMENT 1, AND KE IS SET TO "1" AND CONTROL IS RETURNED TO BOXPLX.

THE SECOND FUNCTION THE USER MUST PROVIDE EVALUATES THE OBJECTIVE FUNCTION. IT IS CALLED FE(X) AS SHOWN IN THE EXAMPLE ABOVE, AND FE MUST BE SET TO THE VALUE OF THE OBJECTIVE FUNCTION CORRESPONDING TO CURRENT VALUES OF THE NV INDEPENDENT VARIABLES IN ARRAY 'X'.

REFERENCES

BOX, M. J., "A NEW METHOD OF CONSTRAINED OPTIMIZATION AND A


```

      VT = BU(I)
      2 IF (NPR.GT.0) WRITE (6,49) II
      3 V(I,1) = VT
        CEN(I) = VT
        IF (IP.EQ.1) GO TO 4
        BL(I) = BL(I)+AMAX1(EP,EP*ABS(BL(I)))
        BU(I) = BU(I)-AMAX1(EP,EP*ABS(BU(I)))
      4 SUM(I) = VT
        C
      NCE = 1 NUMBER OF CONSTRAINT EVALUATIONS
      I = 1
      IF (KE(V(1,1)).EQ.0) GO TO 5
      IF (NPR.LE.0) GO TO 12
      WRITE (6,50)
      GO TO 12
      5 NFE = 1
        C
      NUMBER OF VERTICES (K) = 2 TIMES NO. OF VARIABLES.
      K = 2*NV
        C
      NUMBER OF DISPLACEMENTS ALLOWED.
      NLIM = 5*NV+10
        C
      NUMBER OF CONSECUTIVE TRIALS WITH UNCHANGED FE TO TERMINATE.
      NCT = NLIM+NV
      ALPHA = 1.3
      FK = K
      FKM = FK-1.
      BETA = ALPHA+1.
        C
      INSURE SEED OF RANDOM NUMBER GENERATOR IS ODD.
      IQR = R*1.E7
      IF (MOD(IQR,2).EQ.0) IQR=IQR+101
        C
        SET UP INITIAL VERTICES
      FUN(1) = FE(V(1,1))
      YMN = FUN(1)
      6 FI = 1.
      FUNOLD = FUN(1)
        C
      DO 15 I=2,K
      FI = FI+1.
      LIMIT = 0
      7 LIMIT = LIMIT+1
        C
      END CALCULATION IF FEASIBLE CENTROID CANNOT BE FOUND.

```

```

BXPX3320
BXPX3330
BXPX3340
BXPX3350
BXPX3360
BXPX3370
BXPX3380
BXPX3390
BXPX3400
BXPX3410
BXPX3420
BXPX3430
BXPX3440
BXPX3450
BXPX3460
BXPX3470
BXPX3480
BXPX3490
BXPX3500
BXPX3510
BXPX3520
BXPX3530
BXPX3540
BXPX3550
BXPX3560
BXPX3570
BXPX3580
BXPX3590
BXPX3600
BXPX3610
BXPX3620
BXPX3630
BXPX3640
BXPX3650
BXPX3660
BXPX3670
BXPX3680
BXPX3690
BXPX3700
BXPX3710
BXPX3720
BXPX3730
BXPX3740
BXPX3750
BXPX3760
BXPX3770
BXPX3780
BXPX3790

```


BXPX3800
 BXPX3810
 BXPX3820
 BXPX3830
 BXPX3840
 BXPX3850
 BXPX3860
 BXPX3870
 BXPX3880
 BXPX3890
 BXPX3900
 BXPX3910
 BXPX3920
 BXPX3930
 BXPX3940
 BXPX3950
 BXPX3960
 BXPX3970
 BXPX3980
 BXPX3990
 BXPX4000
 BXPX4010
 BXPX4020
 BXPX4030
 BXPX4040
 BXPX4050
 BXPX4060
 BXPX4070
 BXPX4080
 BXPX4090
 BXPX4100
 BXPX4110
 BXPX4120
 BXPX4130
 BXPX4140
 BXPX4150
 BXPX4160
 BXPX4170
 BXPX4180
 BXPX4190
 BXPX4200
 BXPX4210
 BXPX4220
 BXPX4230
 BXPX4240
 BXPX4250
 BXPX4260
 BXPX4270

```

C      IF (LIMIT,GE,NLIM) GO TO 11
C      DO 8 J=1,NV
C      RANDOM NUMBER GENERATOR (RANDU)
      IQR = IQR*65539
      IF (IQR.LT.0) IQR = IQR+2147483647+1
      RQX = IQR
      RQX = RQX*.4656613E-9
      V(J,I) = BL(J)+RQX*(BU(J)-BL(J))
      IF (IP.EQ.1) V(J,I)=AINT(V(J,I)+.5)
8     CONTINUE
C
      DO 10 L=1,NLIM
      NCE = NCE+1
      IF (KE(V(1,I)).EQ.0) GO TO 13
C
      DO 9 J=1,NV
      VT = .5*(V(J,I)+CEN(J))
      IF (IP.EQ.1) VT = AINT(VT+.5)
      V(J,I) = VT
9     CONTINUE
C
10    CONTINUE
C
11    IF (NPR.LE.0) GO TO 12
      WRITE (6,51) I
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,I,FUN,CEN,I)
12    IER = -1
      GO TO 48
C
13    DO 14 J=1,NV
      SUM(J) = SUM(J)+V(J,I)
14    CEN(J) = SUM(J)/FI
C
      TRY TO ASSURE FEASIBLE CENTROID FOR STARTING.
      NCE = NCE+1
      IF (KE(CEN).NE.0) GO TO 7
      NFE = NFE+1
      FUN(I) = FE(V(1,I))
15    CONTINUE
C
      END OF LOOP SETTING OF INITIAL COMPLEX.
      IF (NPR.LE.0) GO TO 17
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,0)
C
      FIND THE WORST VERTEX, THE 'J'TH.
      J = 1
  
```



```

C      DO 16 I=2,K
C      IF (FUN(J).GE.FUN(I)) GO TO 16
C      J=I
C      16 CONTINUE
C      BASIC LOOP.  ELIMINATE EACH WORST VERTEX IN TURN.  IT MUST BECOME
C      NO LONGER WORST, NOT MERELY IMPROVED.  FIND NEXT-TO-WORST VERTEX,
C      THE JN,TH ONE.
C      17 JN = 1
C      IF (J.EQ.1) JN = 2
C      DO 18 I=1,K
C      IF (I.EQ.J) GO TO 18
C      IF (FUN(JN).GE.FUN(I)) GO TO 18
C      JN = I
C      18 CONTINUE
C      LIMIT = NUMBER OF MOVES DURING THIS TRIAL TOWARD THE CENTROID
C      DUE TO FUNCTION VALUE.
C      LIMIT = 1
C      COMPUTE CENTROID AND OVER REFLECT WORST VERTEX.
C      DO 19 I=1,NV
C      VT = V(I,J)
C      SUM(I) = SUM(I)-VT
C      CEN(I) = SUM(I)/FKM
C      VT = BETA*CEN(I)-ALPHA*VT
C      IF (IP.EQ.1) VT = AINT(VT+.5)
C      INSURE THE EXPLICIT CONSTRAINTS ARE OBSERVED.
C      19 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))
C      NT = NT+1
C      CHECK FOR IMPLICIT CONSTRAINT VIOLATION.
C      20 DO 25 N=1,NLIM
C      NCE = NCE+1
C      IF (KE(V(1,J)).EQ.0) GO TO 26
C      EVERY KV,TH TIME, OVER-REFLECT THE OFFENDING VERTEX THROUGH THE
C      BEST VERTEX.
C      IF (MOD(N,KV).NE.0) GO TO 22
C      CALL FBV(K,FUN,M)
C      DO 21 I=1,NV

```

```

BXPX4280
BXPX4290
BXPX4300
BXPX4310
BXPX4320
BXPX4330
BXPX4340
BXPX4350
BXPX4360
BXPX4370
BXPX4380
BXPX4390
BXPX4400
BXPX4410
BXPX4420
BXPX4430
BXPX4440
BXPX4450
BXPX4460
BXPX4470
BXPX4480
BXPX4490
BXPX4500
BXPX4510
BXPX4520
BXPX4530
BXPX4540
BXPX4550
BXPX4560
BXPX4570
BXPX4580
BXPX4590
BXPX4600
BXPX4610
BXPX4620
BXPX4630
BXPX4640
BXPX4650
BXPX4660
BXPX4670
BXPX4680
BXPX4690
BXPX4700
BXPX4710
BXPX4720
BXPX4730
BXPX4740
BXPX4750

```



```

VT = BETA*V(I,M)-ALPHA*V(I,J)
IF (IP.EQ.1) VT = AINT(VT+.5)
21 V(I,J) = AMAX1(AMINI(VT,BU(I)),BL(I))
C
GO TO 24
C
CONSTRAINT VIOLATION: MOVE NEW POINT TOWARD CENTROID.
C
22 DO 23 I=1,NV
VT = .5*(CEN(I)+V(I,J))
IF (IP.EQ.1) VT = AINT(VT+.5)
V(I,J) = VT
23 CONTINUE
C
24 NT = NT+1
25 CONTINUE
C
IER = 1
C
CANNOT GET FEASIBLE VERTEX BY MOVING TOWARD CENTROID,
OR BY OVER-REFLECTING THRU THE BEST VERTEX.
IF (NPR.LE.0) GO TO 42
WRITE (6,52) NT,J
CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FJN,CEN,J)
GO TO 42
C
FEASIBLE VERTEX FOUND, EVALUATE THE OBJECTIVE FUNCTION.
26 NFE=NFE+1
FUNTRY = FE(V(1,J))
C
TEST TO SEE IF FUNCTION VALUE HAS NOT CHANGED.
AFO = ABS(FUNTRY-FUNOLD)
AMX = AMAX1(ABS(EP*FUNOLD),EP)
C
ACTIVATE THE FOLLOWING TWO STATEMENTS FOR DIAGNOSTIC PURPOSES ONLY.
WRITE (6,99) J,AFO,AMX,FUNTRY,FUNOLD,FUN(J),FUN(JN),NTFS,N
99 FORMAT (1X,I3,6E15.7,2I5)
IF (AFO.GT.AMX) GO TO 27
NTFS = NTFS+1
IF (NTFS.LT.NCT) GO TO 28
IER = 0
IF (NPR.LE.0) GO TO 42
WRITE (6,53) K
GO TO 42
27 NTFS = 0
C
IS THE NEW VERTEX NO LONGER WORST?
28 IF (FUNTRY.LT.FUN(JN)) GO TO 34
C

```



```

C TRIAL VERTEX IS STILL WORST; ADJUST TOWARD CENTROID.
C EVERY KV,TH TIME, OVER-REFLECT THE OFFENDING VERTEX THROUGH THE
C BEST VERTEX.
C   LIMIT = LIMIT+1
C   IF (MOD(LIMIT,KV).NE.0) GO TO 30
C   CALL FBV (K,FUN,M)
C
C   DO 29 I=1,NV
C   VT = BETA*V(I,M) - ALPHA*V(I,J)
C   IF (IP.EQ.1) VT = AINT(VT+.5)
C   29 V(I,J) = AMAX1(AMIN1(VT,BU(I)),BL(I))
C
C   GO TO 32
C
C   DO 30 I=1,NV
C   VT = .5*(CEN(I)+V(I,J))
C   IF (IP.EQ.1) VT = AINT(VT+.5)
C   30 V(I,J) = VT
C   31 CONTINUE
C
C   32 IF (LIMIT.LT.NLIM) GO TO 33
C
C   CANNOT MAKE THE J,TH VERTEX NO LONGER WORST BY DISPLACING TOWARD
C   THE CENTROID OR BY OVER-REFLECTING THRU THE BEST VERTEX.
C   IER = 2
C   IF (NPR.GT.0) WRITE (6,52) NT,J
C   GO TO 42
C   33 NT = NT+1
C   GO TO 20
C
C   SUCCESS: WE HAVE A REPLACEMENT FOR VERTEX J.
C   34 FUN(J) = FUNTRY
C   FUNOLD = FUNTRY
C   NPT = NPT+1
C
C   EVERY 100,TH PERMISSIBLE TRIAL, RECOMPUTE CENTROID SUMMATION TO
C   AVOID CREEPING ERROR.
C   IF (MOD(NPT,100).NE.0) GO TO 37
C
C   DO 36 I=1,NV
C   SUM(I) = 0.
C
C   DO 35 N=1,K
C   SUM(I) = SUM(I)+V(I,N)
C
C   CEN(I) = SUM(I)/FK
C   36 CONTINUE

```

```

BXPX5240
BXPX5250
BXPX5260
BXPX5270
BXPX5280
BXPX5290
BXPX5300
BXPX5310
BXPX5320
BXPX5330
BXPX5340
BXPX5350
BXPX5360
BXPX5370
BXPX5380
BXPX5390
BXPX5400
BXPX5410
BXPX5420
BXPX5430
BXPX5440
BXPX5450
BXPX5460
BXPX5470
BXPX5480
BXPX5490
BXPX5500
BXPX5510
BXPX5520
BXPX5530
BXPX5540
BXPX5550
BXPX5560
BXPX5570
BXPX5580
BXPX5590
BXPX5600
BXPX5610
BXPX5620
BXPX5630
BXPX5640
BXPX5650
BXPX5660
BXPX5670
BXPX5680
BXPX5690
BXPX5700
BXPX5710

```


BXPX5720
 BXPX5730
 BXPX5740
 BXPX5750
 BXPX5760
 BXPX5770
 BXPX5780
 BXPX5790
 BXPX5800
 BXPX5810
 BXPX5820
 BXPX5830
 BXPX5840
 BXPX5850
 BXPX5860
 BXPX5870
 BXPX5880
 BXPX5890
 BXPX5900
 BXPX5910
 BXPX5920
 BXPX5930
 BXPX5940
 BXPX5950
 BXPX5960
 BXPX5970
 BXPX5980
 BXPX5990
 BXPX6000
 BXPX6010
 BXPX6020
 BXPX6030
 BXPX6040
 BXPX6050
 BXPX6060
 BXPX6070
 BXPX6080
 BXPX6090
 BXPX6100
 BXPX6110
 BXPX6120
 BXPX6130
 BXPX6140
 BXPX6150
 BXPX6160
 BXPX6170
 BXPX6180
 BXPX6190

```

C      LC = 0
C      GO TO 39
C
C      37 DO 38 I=1,NV
C      38 SUM(I) = SUM(I)+V(I,J)
C
C      LC = J
C
C      39 IF (NPR.LE.0) GO TO 40
C      IF (MOD(NPT,NPR).NE.0) GO TO 40
C
C      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,CEN,LC)
C
C      HAS THE MAX. NUMBER OF TRIALS BEEN REACHED WITHOUT CONVERGENCE?
C      IF NOT, GO TO NEW TRIAL.
C      40 IF (NT.GE.NTA) GO TO 41
C
C      NEXT-TO-WORST VERTEX NOW BECOMES WORST.
C      J = JN
C      GO TO 17
C      41 IER = 3
C      IF (NPR.GT.0) WRITE (6,54)
C
C      COLLECTOR POINT FOR ALL ENDINGS.
C      1) CANNOT DEVELOP FEASIBLE VERTEX.
C      2) CANNOT DEVELOP A NO-LONGER-WORST VERTEX.
C      3) FUNCTION VALUE UNCHANGED FOR K TRIALS.
C      4) LIMIT ON TRIALS REACHED.
C      5) CANNOT FIND FEASIBLE VERTEX AT START.
C
C      IER = 1
C      IER = 2
C      IER = 0
C      IER = 3
C      IER = -1
C
C      42 CONTINUE
C
C      FIND BEST VERTEX.
C      CALL FBV (K,FUN,M)
C      IF (IER.GE.3) GO TO 44
C
C      RESTART IF THIS SOLUTION IS SIGNIFICANTLY BETTER THAN THE PREVIOUS,
C      OR IF THIS IS THE FIRST TRY.
C      IF (NPR.LE.0) GO TO 43
C      WRITE (6,55) (M,YMN,FUN(M))
C      43 IF (FUN(M).GE.YMN) GO TO 47
C      IF (ABS(FUN(M)-YMN).LE.AMAX1(EP,EP*YMN)) GO TO 47
C
C      GIVE IT ANOTHER TRY UNLESS LIMIT ON TRIALS REACHED.
C      44 YMN = FUN(M)
C      FUN(1) = FUN(M)
C
C      DO 45 I=1,NV
  
```



```

C      CEN(I) = V(I,M)
      SUM(I) = V(I,M)
45  V(I,1) = V(I,M)
C
C      DO 46 I=1,NVT
46  XS(I) = V(I,M)
C
C      IF (IER.LT.3) GO TO 6
47  IF (NPR.LE.0) GO TO 48
      CALL BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FUN,V(1,M),-1)
48  WRITE (6,56) FUN(M)
      RETURN
C
49  FORMAT (50H) INDEX AND DIRECTION OF OUTLYING VARIABLE AT START(I5)
50  FORMAT (50H) IMPLICIT CONSTRAINT VIOLATED AT START. DEAD END. )
51  FORMAT (50H) CANNOT FIND FEASIBLE, I4, TH VERTEX OR CENTROID AT START
      I.,.)
52  FORMAT (10H) AT TRIAL I4,54H CANNOT FIND FEASIBLE VERTEX WHICH IS
      NO LONGER WORST, I4,15X, RESTART FROM BEST VERTEX.,)
53  FORMAT (40H) FUNCTION HAS BEEN ALMOST UNCHANGED FOR I5,7H TRIALS)
54  FORMAT (27H) LIMIT ON TRIALS EXCEEDED.,)
55  FORMAT (50H) BEST VERTEX IS NO., I3, OLD MIN WAS , E15.7,
      I., NEW MIN IS , E15.7)
56  FORMAT (50H) MIN OBJECTIVE FUNCTION IS , E15.7)
      END
      SUBROUTINE FBV (K,FUN,M)
      DIMENSION FUN(50)
      M = 1
C
      DO 1 I=2,K
      IF (FUN(M).LE.FUN(I)) GO TO 1
      M = I
1  CONTINUE
C
      RETURN
      END
      SUBROUTINE BOUT (NT,NPT,NFE,NCE,NV,NVT,V,K,FN,C,IK)
      DIMENSION V(50,50), FN(50), C(25)
      WRITE (6,4) NT,NPT,NFE,NCE
C
      DO 1 I=1,K
      WRITE (6,5) FN(I), (V(J,I), J=1,NV)
      IF (NVT.LE.NV) GO TO 1
      NVP = NV+1
      WRITE (6,6) (V(J,I), J=NVP,NVT)
1  CONTINUE
C
      IF (IK.NE.0) GO TO 2

```

```

BX PX 6200
BX PX 6210
BX PX 6220
BX PX 6230
BX PX 6240
BX PX 6250
BX PX 6260
BX PX 6270
BX PX 6280
BX PX 6290
BX PX 6300
BX PX 6310
BX PX 6320
BX PX 6330
BX PX 6340
BX PX 6350
BX PX 6360
BX PX 6370
BX PX 6380
BX PX 6390
BX PX 6400
BX PX 6410
BX PX 6420
BX PX 6430
BX PX 6440
BX PX 6450
BX PX 6460
BX PX 6470
BX PX 6480
BX PX 6490
BX PX 6500
BX PX 6510
BX PX 6520
BX PX 6530
BX PX 6540
BX PX 6550
BX PX 6560
BX PX 6570
BX PX 6580
BX PX 6590
BX PX 6600
BX PX 6610
BX PX 6620
BX PX 6630
BX PX 6640
BX PX 6650
BX PX 6660
BX PX 6670

```


BXPX6680
 BXPX6690
 BXPX6700
 BXPX6710
 BXPX6720
 BXPX6730
 BXPX6740
 BXPX6750
 BXPX6760
 BXPX6770
 BXPX6780
 BXPX6790
 BXPX6800
 BXPX6820
 BXPX6830
 BXPX6840
 BXPX6850
 BXPX6860

```

WRITE (6,7) (C(I), I=1,NV)
RETURN
2 IF (IK,GE,0) GO TO 3
WRITE (6,8) (C(I), I=1,NV)
RETURN
3 WRITE (6,9) IK, (C(I), I=1,NV)
RETURN

```

[illegible]

۷

5

FUNCTION DEGRAD

This function was programmed to convert degrees to radians and radians to degrees. A third purpose is to convert degrees to a range of 0-360. It is used extensively throughout the programs, functions, and subroutines listed in this thesis.

FUNCTION DEGRAD

```

C C C C C C C C C C C C C C C C
FUNCTION DEGRAD (NDRFLG,NSHFLG,FUNCT)
FUNCTION TO CONVERT DEGREES TO RADIAN AND RADIANS TO
DEGREES AND SHIFT THE DEGREES TO A RANGE BETWEEN 000.0
AND 360.0 IN ACCORDANCE WITH THE FOLLOWING FLAG
DEFINITIONS
NDRFLG 0 - CONVERT RADIAN TO DEGREES
        1 - CONVERT DEGREES TO RADIAN
        2 - DO NOT CONVERT
NSHFLG 0 - CONVERT ANGLES(DEG) TO THE RANGE BETWEEN 000.0
        1 - DO NOT CONVERT
        AND 360.0
        1 - DO NOT CONVERT
FUNCU = FUNCT
IF (NDRFLG.EQ.2) GO TO 2
IF (NDRFLG.EQ.1) GO TO 1
FUNCU = FUNCU*180.0/3.141592654
GO TO 2
1 FUNCU = FUNCU*3.141592654/180.0
GO TO 5
2 IF (NSHFLG.EQ.1) GO TO 5
3 IF (FUNCU.LT.360.0) GO TO 4
FUNCU = FUNCU-360.0
GO TO 3
4 IF (FUNCU.GE.0.0) GO TO 5
FUNCU = FUNCU+360.0
GO TO 4
5 DEGRAD = FUNCU
RETURN
END
DGRD 10
DGRD 20
DGRD 30
DGRD 40
DGRD 50
DGRD 60
DGRD 70
DGRD 80
DGRD 90
DGRD 100
DGRD 110
DGRD 120
DGRD 130
DGRD 140
DGRD 150
DGRD 160
DGRD 170
DGRD 180
DGRD 190
DGRD 200
DGRD 210
DGRD 220
DGRD 230
DGRD 240
DGRD 250
DGRD 260
DGRD 270
DGRD 280
DGRD 290
DGRD 300
DGRD 310
DGRD 320

```


FUNCTION DELAY

This function is used as the time delay in the speed control optimization runs. It was designed to be used as an equivalence to DELY in DSL simulation. The following variables are defined:

E(I) is the storage array (should be initialized before the first function call)
K is the delay step count
SPDDES is the variable to be delayed for K steps
P is the flag for delay or no delay
 $P \geq 0.0$ delay SPDDES
 $P < 0.0$ function output equal to SPDDES

The function stores the input value (SPDDES) in E(M) and decrements the value in array E(I) at each call of the function until the value is in the position of E(1). The value is then output from the function delayed K intervals.

FUNCTION DELAY

```

C
FUNCTION DELAY (K,P,SPDDES,E)
DIMENSION E(10)
M = K+1
E(M) = SPDDES
DO 1 I=1,K
1 E(I) = E(I+1)
IF (P.LT.0.0) GO TO 2
DELAY = E(1)
RETURN
2 DELAY = SPDDES
RETURN
END
C

```

```

DELY 10
DELY 20
DELY 30
DELY 40
DELY 50
DELY 60
DELY 70
DELY 80
DELY 90
DELY 100
DELY 110
DELY 120
DELY 130
DELY 140

```


FUNCTION FE - RUN A

FEA

This function is the simulation for heading control optimization of the approach phase. It is called by subroutine EOXP LX. The integration step size is 0.04 with a final time of 20.0. In this function all initial conditions are set to zero except initial geographic location and speed. The reference ship maintains a straight course and the control ship starts its approach 5 ship lengths astern and 0.4 ship lengths laterally displaced to starboard of the reference ship.

The function is referred to as function FEA in the text.

FUNCTION FE - RUN A

```

C      FUNCTION FE (Z)
C      EVALUATION OF COST FUNCTION AS A FUNCTION OF RSENS,WTSENS,RGN
C      DIMENSION Z(8),Y(20),YDOT(20),X(20),XDOT(20)
C      REAL *8XDOT,XDOT,TD
C      HYDRODYNAMIC COEFFICIENTS
      A11 = 0.015
      B11 = 0.01243
      A21 = 0.00027
      B21 = 0.0051
      A12 = 0.000197
      B12 = 0.00351
      A22 = 0.00068
      B22 = 0.00227
      A33 = 0.0085
      B33 = 0.0012
      XKA = 0.0027
      XKB = -0.00126
      XNC = 0.0012
      D = A11*A22-A21*A12
      XKC = 0.0
      XLUC = 20.84765
      DLIDM = 2.0
      DLTEM = 7.0
      XKG = DLIDM/DLTEM
      D2D = 0.0
C      IDENTIFICATION OF GAINS TO BE FOUND
      RSENS = Z(1)
      WTSENS = Z(2)
      RGN = Z(3)
      VFBG = 0.0
C      INITIAL CONDITIONS
      DO 1 J=1,14
1    Y(J) = 0.0
C      INITIAL GEOGRAPHIC LOCATION
      Y(5) = 5.0
      Y(6) = 0.0
      Y(10) = 0.0
      Y(11) = 0.4
C      OTHER INITIALIZATIONS
      N = 1
      RD = 1.0
FEZA 10
FEZA 20
FEZA 30
FEZA 40
FEZA 50
FEZA 60
FEZA 70
FEZA 80
FEZA 90
FEZA 100
FEZA 110
FEZA 120
FEZA 130
FEZA 140
FEZA 150
FEZA 160
FEZA 170
FEZA 180
FEZA 190
FEZA 200
FEZA 210
FEZA 220
FEZA 230
FEZA 240
FEZA 250
FEZA 260
FEZA 270
FEZA 280
FEZA 290
FEZA 300
FEZA 310
FEZA 320
FEZA 330
FEZA 340
FEZA 350
FEZA 360
FEZA 370
FEZA 380
FEZA 390
FEZA 400
FEZA 410
FEZA 420
FEZA 430

```


FEZA 440
FEZA 450
FEZA 460
FEZA 470
FEZA 480
FEZA 490
FEZA 500
FEZA 510
FEZA 520
FEZA 530
FEZA 540
FEZA 550
FEZA 560
FEZA 570
FEZA 580
FEZA 590
FEZA 600
FEZA 610
FEZA 620
FEZA 630
FEZA 640
FEZA 650
FEZA 660
FEZA 670
FEZA 680
FEZA 690
FEZA 700
FEZA 710
FEZA 720
FEZA 730
FEZA 740
FEZA 750
FEZA 760
FEZA 770
FEZA 780
FEZA 790
FEZA 800
FEZA 810
FEZA 820
FEZA 830
FEZA 840
FEZA 850
FEZA 860
FEZA 870
FEZA 880
FEZA 890
FEZA 900
FEZA 910

```

IS = 1
DD = 0.2
DI = 0.0
D2 = 0.0
YY1 = 0.0
YY2 = 0.0
YN1 = 0.0
YN2 = 0.0
SPEED = 0.0
INITIALIZATIONS
Y(4) = 1.0
U02 = 1.5
CDO12 = 1.5
DISTANCE = 1.5
DYO = Y(10)-Y(5)
DX0 = Y(11)-Y(6)
CALL TRANS (Y(2), DX0, DYO, ADX, ADY)
CALL SLOPES (ADX, ADY, YY1, YY2, YN1, YN2)
INITIALIZE TIME
T = 0.0
DT = 0.04
JT = 0
SHIP A
XIF11 = XKA*D1
XIF21 = XKB*D1
XIF31 = XKC*D1 + XNC
XI11 = -B11*Y(1) - B21*Y(3) + XIF11
XI21 = -B12*Y(1) - B22*Y(3) + XIF21
XI31 = -B31*Y(4) + XIF31
YDOT(1) = (XI11*A22 - XI21*A21)/D
YDOT(2) = Y(3)
YDOT(3) = (XI21*A11 - XI11*A12)/D
YDOT(4) = XI31/A33
YDOT(5) = Y(4)*COS(Y(2)) - Y(1)*SIN(Y(2))
YDOT(6) = Y(4)*SIN(Y(2)) + Y(1)*COS(Y(2))
SHIP B
XIF12 = XKA*D2 + YY2
XIF22 = XKB*D2 + YN2
XIF32 = XKC*D2 + XNC
XI12 = -B11*Y(7) - B21*Y(9) + XIF12
XI22 = -B12*Y(7) - B22*Y(9) + XIF22
CDO12 = -SPDCR(ADX, U02)
XI32 = -B33*CDO12 + XIF32
YDOT(7) = (XI12*A22 - XI22*A21)/D
YDOT(8) = Y(9)
YDOT(9) = (XI22*A11 - XI12*A12)/D
YDOT(10) = CDO12*COS(Y(8)) - Y(7)*SIN(Y(8))
YDOT(11) = CDO12*SIN(Y(8)) + Y(7)*COS(Y(8))
DX = Y(10) - Y(5)

```



```

DY = Y(11)-Y(6)
CALL TRANS (Y(2),DX,DY,ADX,ADY)
CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)
YAWD2 = DEGRAD(0,0,Y(8))
CALL RBMEAS (N,Y(2),Y(5),Y(6),Y(8),Y(10),Y(11),RD,R1,B1,BB1,R2,B2,
1BB2)
CALL HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,Y(8),PSIDFD,PSIADD,PSI
1DED,WT,DA,AID,B1D,B2D,WISENS,DD,RD)
BCOT2D = DEGRAD(0,1,Y(9))
BDOITFB = VFBG*BDOIT2D
DDUMB = YAWD2-PSIDED+BDOITFB
IF (DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF (DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS = XLIMIT(-30.0,30.0,DDUMB*RGJ)
DLTE = DLTS-D2D
DLTBE = XLIMIT(-DLTEM,DLTEM,DLTE)
YDOT(14) = XKG*DLTBE*XLUC
D2D = Y(14)
D2 = DEGRAD(1,1,D2D)
DTRAN = T*ABS(D2)
YDOT(12) = DTRAN
DISTE = T*10.0*ABS(DD-ADY)
YDOT(13) = DISTE
OBJ = Y(12)+Y(13)

DO 3 J=1,14
X(J) = DBLE(Y(J))
3 XDOT(J) = DBLE(YDOT(J))

TD = DBLE(T)
DTD = DBLE(DT)
ZS = RKLDEQ(14,X,XDOT,TD,DTD,JT)

DO 4 J=1,14
Y(J) = SNGL(X(J))
4 YDOT(J) = SNGL(XDOT(J))

T = SNGL(TD)
DT = SNGL(DTD)
IF (ZS-1.) 5,2,6
5 WRITE (6,8)
STOP
6 IF (T.GT.20.0) GO TO 7
GO TO 2
7 FE = OBJ
WRITE (6,9) OBJ,RSENS,WTSENS,RGJ
RETURN

```

```

FEZA 920
FEZA 930
FEZA 940
FEZA 950
FEZA 960
FEZA 970
FEZA 980
FEZA 990
FEZA1000
FEZA1010
FEZA1020
FEZA1030
FEZA1040
FEZA1050
FEZA1060
FEZA1070
FEZA1080
FEZA1090
FEZA1100
FEZA1110
FEZA1120
FEZA1130
FEZA1140
FEZA1150
FEZA1160
FEZA1170
FEZA1180
FEZA1190
FEZA1200
FEZA1210
FEZA1220
FEZA1230
FEZA1240
FEZA1250
FEZA1260
FEZA1270
FEZA1280
FEZA1290
FEZA1300
FEZA1310
FEZA1320
FEZA1330
FEZA1340
FEZA1350
FEZA1360
FEZA1370
FEZA1380
FEZA1390

```




FEZAI400
FEZAI410
FEZAI420
FEZAI430

```
8 FORMAT (' RKLDEQ RETURNED VALUE LT 1.0, INTEGRATION PROBLEM')  
9 FORMAT (' EXIT FUNCTION FE(Z) OBJ= ', F15.8, ' RSENS= ', F15.8, '  
1 WTSENS= ', F15.8, ' RGN= ', F15.8)  
END
```


FUNCTION FE - RUN B
FEB

This function is the simulation for heading control optimization of the turn phase. It is called by subroutine BOXPLX. The integration step size is 0.04 with a final time of 20.0. In this function, the following initial conditions are non-zero:

control ship rudder angle D2D & Y(14) = 8.7 degrees
lateral displacement Y(11) = 0.2
reference ship's speed U01 & Y(4) = 1.0
control ship's speed U02 & CDCT2 = 1.5 (after first
step becomes 1.0)

The reference ship's rudder is activated to 5.0 degrees between time 4.0 and 5.0. The runs were for port side replenishment.

The function is referred to as function FEB in the text.


```
C      FUNCTION FE (Z)
      EVALUATION OF COST FUNCTION AS A FUNCTION OF RSENS,WTSENS,RGN
      DIMENSION Z(8),Y(20),YDOT(20),X(20),XDOT(20)
      REAL *8XDOT,X,DT0,TD
      HYDRODYNAMIC COEFFICIENTS
      A11 = 0.015
      B11 = 0.01243
      A21 = 0.00027
      B21 = 0.0051
      A12 = 0.000197
      B12 = 0.00351
      A22 = 0.00068
      B22 = 0.00227
      A33 = 0.0085
      B33 = 0.0012
      XK A = 0.0027
      XK B = -0.00125
      XNC = 0.0012
      D = A11*A22-A21*A12
      XK C = 0.0
      XLUC = 20.84765
      DLTEM = 2.0
      DLTEM = 7.0
      XK G = DLDM/DLTEM
      D10 = 0.0
      D2D = 8.7
      IDENTIFICATION OF GAINS TO BE FOUND
      RSENS = Z(1)
      WTSENS = Z(2)
      RGN = Z(3)
      VFBG = Z(4)
      INITIAL CONDITIONS
      DO 1 J=1,14
1     Y(J) = 0.0
      INITIAL GEOGRAPHIC LOCATION
      Y(5) = 0.0
      Y(6) = 0.0
      Y(10) = 0.0
      Y(11) = 0.2
      Y(14) = 8.7
      OTHER INITIALIZATIONS
```

FEZB 10
FEZB 20
FEZB 30
FEZB 40
FEZB 50
FEZB 60
FEZB 70
FEZB 80
FEZB 90
FEZB 100
FEZB 110
FEZB 120
FEZB 130
FEZB 140
FEZB 150
FEZB 160
FEZB 170
FEZB 180
FEZB 190
FEZB 200
FEZB 210
FEZB 220
FEZB 230
FEZB 240
FEZB 250
FEZB 260
FEZB 270
FEZB 280
FEZB 290
FEZB 300
FEZB 310
FEZB 320
FEZB 330
FEZB 340
FEZB 350
FEZB 360
FEZB 370
FEZB 380
FEZB 390
FEZB 400
FEZB 410
FEZB 420
FEZB 430

FEZB 440
FEZB 450
FEZB 460
FEZB 470
FEZB 480
FEZB 490
FEZB 500
FEZB 510
FEZB 520
FEZB 530
FEZB 540
FEZB 550
FEZB 560
FEZB 570
FEZB 580
FEZB 590
FEZB 600
FEZB 610
FEZB 620
FEZB 630
FEZB 640
FEZB 650
FEZB 660
FEZB 670
FEZB 680
FEZB 690
FEZB 700
FEZB 710
FEZB 720
FEZB 730
FEZB 740
FEZB 750
FEZB 760
FEZB 770
FEZB 780
FEZB 790
FEZB 800
FEZB 810
FEZB 820
FEZB 830
FEZB 840
FEZB 850
FEZB 860
FEZB 870
FEZB 880
FEZB 890
FEZB 900
FEZB 910

```

N = 1
RD = 1.0
IS = 1.0
DD = 0.2
D1 = 0.0
D2 = DEGRAD(1,1,02D)
YY1 = 0.0
YY2 = 0.0
YN1 = 0.0
YN2 = 0.0
SPEED INITIALIZATIONS
Y(4) = 1.0
U01 = 1.0
U02 = 1.5
CDOI2 = 1.5
DISTANCE INITIALIZATION
DY0 = Y(10)-Y(5)
DX0 = Y(11)-Y(6)
CALL TRANS (Y(2),DX0,DY0,ADX,ADY)
CALL SLOPES (ADX,ADY,YY1,YY2,YN1,YN2)
INITIALIZE TIME
T = 0.0
DT = 0.04
JT = 0
SHIP A
2 DIDES = 0.0
IF ((T-GE.4.0).AND.(T.LE.5.0)) DIDES=5.0
DLTSL = XLIMIT(-30.0,30.0,DIDES)
DLTSL-DID
DLTBE1 = XLIMIT(-DLTEM,DLTEM,DLTSL)
YDOT(12) = XKG*DLTBE1*XLUC
Y(12)
DID = DEGRAD(1,1,DID)
D1 = DEGRAD(1,1,D1D)
XIF11 = XKA*D1
XIF21 = XKB*D1
XIF31 = XKC*D1 +XNC
XI11 = -B11*Y(1)-B22*Y(3)+XIF11
XI21 = -B12*Y(1)-B22*Y(3)+XIF21
XI31 = -B33*Y(4)+XIF31
YDOT(1) = (XI11*A22-XI21*A21)/D
YDOT(2) = Y(3)
YDOT(3) = (XI21*A11-XI11*A12)/D
YDOT(4) = XI31/A33
YDOT(5) = Y(4)*COS(Y(2))-Y(1)*SIN(Y(2))
YDOT(6) = Y(4)*SIN(Y(2))+Y(1)*COS(Y(2))
YAWD1 = DEGRAD(0,0,Y(2))
SHIP B
XIF12 = XKA*D2+YY2

```



```

XIF22 = XKB*D2+YN2
XIF32 = XKC*D2+XNC
XI12 = -B11*Y(7)-B21*Y(9)+XIF12
XI22 = -B12*Y(7)-B22*Y(9)+XIF22
XD0T2 = SPDCR(ADX,U01,U02)
XD0T2 = -B33*XD0T2+XIF32
YD0T(7) = (XI12*A22-XI22*A21)/D
YD0T(8) = Y(9)
YD0T(9) = (XI22*A11-XI12*A12)/D
YD0T(10) = CD0T2*COS(Y(8))-Y(7)*SIN(Y(8))
YD0T(11) = CD0T2*SIN(Y(8))+Y(7)*COS(Y(8))
DX = Y(10)-Y(5)
DY = Y(11)-Y(6)
CALL TRANS (Y(2),DX,DY,ADX,ADY)
CALL SLOPES (ADX,ADY,Y(1),Y(2),Y(1),Y(2))
YAWD2 = DEGRAD(0,0,Y(8))
CALL RBMEAS (N,Y(2),Y(5),Y(6),Y(8),Y(10),Y(11),RD,R1,B1,BB1,R2,B2,
1 BB2)
1 CALL HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,Y(8),PSIDFD,PSIADD,PSI
1 DED,WI,DA,AID,B1D,B2D,WISSENS,DD,RD)
BD0T2D = DEGRAD(0,1,Y(9))
BD0TFB = VFBG*BD0T2D
DDUMB = YAWD2-PSIDED+BD0TFB
IF (DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF (DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLIS = XLIMIT(-30.0,30.0,DDUMB*RGN)
DLIE = DLIS-D2D
DLIBE = XLIMIT(-DLIEM,DLIEM,DLIE)
YD0T(14) = XKG*DLTBE*XLUC
D2D = Y(14)
D2 = DEGRAD(1,1,D2D)
DISTIE = ABS(DD-ADY)
YD0T(13) = DISTIE
OBJ = Y(13)

C
D0 3 J=1,14
X(J) = DBLE(Y(J))
3 XD0T(J) = DBLE(YD0T(J))

C
TD = DBLE(T)
DTD = DBLE(DT)
ZS = RKLDEQ(14,X,XD0T,TD,DTD,JT)

C
D0 4 J=1,14
Y(J) = SNGL(X(J))
4 YD0T(J) = SNGL(X(J))

C
T = SNGL(TD)

```

```

FEZB 920
FEZB 930
FEZB 940
FEZB 950
FEZB 960
FEZB 970
FEZB 980
FEZB 990
FEZB 1000
FEZB 1010
FEZB 1020
FEZB 1030
FEZB 1040
FEZB 1050
FEZB 1060
FEZB 1070
FEZB 1080
FEZB 1090
FEZB 1100
FEZB 1110
FEZB 1120
FEZB 1130
FEZB 1140
FEZB 1150
FEZB 1160
FEZB 1170
FEZB 1180
FEZB 1190
FEZB 1200
FEZB 1210
FEZB 1220
FEZB 1230
FEZB 1240
FEZB 1250
FEZB 1260
FEZB 1270
FEZB 1280
FEZB 1290
FEZB 1300
FEZB 1310
FEZB 1320
FEZB 1330
FEZB 1340
FEZB 1350
FEZB 1360
FEZB 1370
FEZB 1380
FEZB 1390

```


FEZBI400
FEZBI410
FEZBI420
FEZBI430
FEZBI440
FEZBI450
FEZBI460
FEZBI470
FEZBI480
FEZBI490
FEZBI500
FEZBI510
FEZBI520
FEZBI530

```

DT = SNGL(DTD)
IF (ZS-1.) 5,2,6
5 WRITE (6,8)
STOP
6 IF (T.GT.20.0) GO TO 7
GO TO 2
7 FE = OBJ
WRITE (6,9) OBJ,ADY,ADX,YAWD1,YAWD2,D2D
RETURN
8 FORMAT (' RKLDEQ RETURNED VALUE LT 1.0; INTEGRATION PROBLEM' )
9 FORMAT (' EXIT FUNCTION FE(Z),OBJ=',F15.8,';ADY=',F15.8,';ADX=',F15.8,';YAWD1=',F15.8,';YAWD2=',F15.8,';D2D=',F15.8)
1 END

```

C

FUNCTION FE - RUN C
FEC

This function is the simplified simulation for speed control optimization of the switching function SW. It is called by subroutine BOXPLX. The function shown is for approach speed of 1.1 and a replenishment speed of 1.0. The runs were made for various realistic combinations to obtain an optimum switching curve.

The run used a step size of 0.04 and a final time of 10.0. The two ships were run linearly with only the longitudinal direction and motion of any concern.

This function is referred to as function FEC in the text.

FUNCTION FE - RUN C

```

C      FUNCTION FE (Z)  X(10), XDOT(10), Y(10), YDOT(10), E(10)
C      DIMENSION Z(8), X(10), XDOT(10), Y(10), YDOT(10), E(10)
C      REAL *8XDOT,X,DID,TD
C      IDENTIFICATION OF GAINS
C      SW=Z(1)
C      INITIAL CONDITIONS
C      N=2
C      APOX = 1.1
C      BPOX = 0.0
C      DO 1 I=1,N
C      Y(I) = 0.0
C      1 YDOT(I) = 0.0
C      DO 2 I=1,10
C      E(I) = 0.0
C      SPDO1 = 1.0
C      SPDO2 = 1.1
C      Y(1) = 1.1
C      XK1 = 1.012425
C      P = -1.0
C      A = 22.0
C      G = 0.092
C      XLUC = 20.84765
C      UF = 21.73
C      XK = UF/A
C      T = 0.0
C      DT = 0.04
C      JT = 0
C      SPDDER = XK1*SPDO1
C      ADX = BPOX-APOX
C      ITERATION LOOP
C      3 IF (T.GT.0.24) P = 4.88/XLUC
C      IF (ABS(ADX).GT.2.0) GO TO 9
C      SPDDER = XK1*(SPDREC(ADX,SPDO1,SPDO2,SW)-SPDO1)
C      SPDDER = DELAY(6,P,SPDDER,E)
C      SPDIN = XK*(SPDDEL+SPDDER)
C      SPDERR = G*(SPDIN-Y(1))
C      YDOT(1) = SPDERR*XLUC
C      YDOT(2) = T*ABS(ADX)
C      OBJ = Y(2)

```



```

C      DO 4 I=1,N
        X(I) = DBLE(Y(I))
        4 XDOT(I) = DBLE(YDOT(I))
C
        TD = DBLE(T)
        DTD = DBLE(DT)
        ZS = RKLDEQ(N,X,XDOT,TD,DTD,JT)
C
        DO 5 I=1,N
            Y(I) = SNGL(X(I))
            5 YDOT(I) = SNGL(XDOT(I))
C
            T = SNGL(TD)
            DT = SNGL(DTD)
            IF (ZS-1.) 6,3,7
            6 WRITE (6,10)
            7 STOP
            IF (T.GT.10.0) GO TO 8
            APOSX = APOSX+SPDOI*DT
            BPOSX = BPOSX+Y(1)*DT
            ADX = BPOSX-APOSX
            GO TO 3
            8 FE = OBJ
              WRITE (6,11) OBJ,SW,ADX,Y(1)
              RETURN
            9 FE = 1.0E06
              WRITE (6,12) ADX
              RETURN
C
            10 FORMAT (' RKLDEQ RETURNED ZS FLAG LT 0.0, INTEGRATION PROBLEM')
            11 FORMAT (' EXIT FUNCTION FE(Z) OBJ=',F15.8,5X,'SW=',F15.8,/,
            12 FORMAT (' EXIT FUNCTION FE(Z) ABS(ADX).GT.2.0, ADX=',F15.8)
            END
FEZC 440
FEZC 450
FEZC 460
FEZC 470
FEZC 480
FEZC 490
FEZC 500
FEZC 510
FEZC 520
FEZC 530
FEZC 540
FEZC 550
FEZC 560
FEZC 570
FEZC 580
FEZC 590
FEZC 600
FEZC 610
FEZC 620
FEZC 630
FEZC 640
FEZC 650
FEZC 660
FEZC 670
FEZC 680
FEZC 690
FEZC 700
FEZC 710
FEZC 720
FEZC 730
FEZC 740
FEZC 750
FEZC 760
FEZC 770
FEZC 780

```


SUBROUTINE HDGRAS

This subrcutine was programmed to calculate the desired heading (FSIDES) for RAS heading control. It uses the outputs of subroutine RBMEAS to calculate this heading with gains RSENS and WTSENS. The large number of outputs in the subroutine call statement were made for ease of DSL printed output for tracking of simulation accuracy.

The subroutine also incorporates a loop to avoid computer precision problems in the ARSIN function.

SUBROUTINE HDGRAS

```

SUBROUTINE HDGRAS (N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,PSIB,PSIDFD,PSIA
1DD,PSIDED,DA,AID,BID,B2D,WTSENS,DD,D) HDGR 10
SUBROUTINE TO CALCULATE DESIRED HEADING FOR RAS HDGR 20
IF N SET TO 1 HDGRAS WILL BE USED HDGR 30
HDGR 40
HDGR 50
HDGR 60
HDGR 70
HDGR 80
HDGR 90
IS DENOTES THE SIDE OF APPROACH OF RECEIVING SHIP IS=1 PORT, HDGR 100
IS=0 STBD HDGR 110
RSENS - ADDITIONAL HEADING SENSITIVITY DUE TO SEPARATION DISTANCE HDGR 120
RSENS=1.0 CORRESPONDS TO 10.87 DEG/10FT FO A 527 FT HDGR 130
SHIP HDGR 140
PSIDFD - ADDITIONAL HEADING DUE TO HEADING DIFFERENCE(WEIGHTED) HDGR 150
PSIADD - ADDITIONAL HEADING DUE TO DISTANCE ERROR(WEIGHTED) HDGR 160
PSIDED - TOTAL DESIRED HEADING=PSIDFD(WEIGHTED)+PSIADD(WEIGHTED) HDGR 170
NOTE ALL RETURNED VALUES ENDING IN "D" ARE IN DEGREES HDGR 180
WTSENS - WEIGHTING FACTOR GAIN FOR DIFFERENCE IN HEADINGS HDGR 190
DA - CG ABS DISTANCE BETWEEN SHIPS HDGR 200
AID - CG REL BEARING BETWEEN SHIPS HDGR 210
HDGR 220
HEADING DIFFERENCE HDGR 230
ARG = (R1*SIN(B1))-R2*SIN(B2))/D HDGR 240
IF (ARG.GT.1.0) GO TO 1 HDGR 250
IF (ARG.LT.-1.0) GO TO 2 HDGR 260
PSIDIF = ARSIN(ARG) HDGR 270
GO TO 3 HDGR 280
1 PSIDIF = ARSIN(1.0) HDGR 290
GO TO 3 HDGR 300
2 PSIDIF = ARSIN(-1.0) HDGR 310
3 CONTINUE HDGR 320
CG DISTANCE HDGR 330
DA = (R1+R2)/2.0 HDGR 340
CG RELATIVE BEARING HDGR 350
AA1 = (BB1+BB2)/2.0 HDGR 360
INITIAL SENSE OF APPROACH SIDE DESIRED DISTANCE HDGR 370
DDC = DD HDGR 380
IF (IS.EQ.0) DDC = -DD HDGR 390
ADDITIONAL HEADING DUE TO DISTANCE HDGR 400
PSIADC = RSENS*(DDC+DA*SIN(AA1)) HDGR 410
TOTAL DESIRED HEADING HDGR 420
PSIDES = PSIADC+WTSENS*PSIDIF+PSIB HDGR 430
CONVERT RADIAN TO DEGREES

```



```

PSIDFD = DEGRAD(0,0,0,PSIDIF)
PSIADD = DEGRAD(0,0,0,PSIADC)
PSIDED = DEGRAD(0,0,0,PSIDES)
A1 = 6.283185307+AA1
A1D = DEGRAD(0,0,A1)
B1D = DEGRAD(0,0,B1)
B2D = DEGRAD(0,0,B2)
4 RETURN
END

```

```

HDGR 440
HDGR 450
HDGR 460
HDGR 470
HDGR 480
HDGR 490
HDGR 500
HDGR 510
HDGR 520

```


FUNCTION KE

This function is required by all optimization runs. It is the function that contains constraints for subrcutine BOXPLX. No constraints are present, consequently function KE is set equal to 0.

FUNCTION KE

FUNCTION KE (X)
DIMENSION X(8)
KE = 0
RETURN
END

KEX
KEX
KEX
KEX
10
20
30
40
50

MAIN PROGRAM FOR FUNCTION MINIMIZATIONS
MINIBXPX

This is a generalized program which calls subrcutine
BOXFLX. Its main purpose is input and output of the values
required in the optimization runs. This is referred to as
MINIBXPX in the text.


```

DIMENSION X(8), XS(8), BU(8), BL(8)
CALL ERRSET (257,256,0,1,1)
READ (5,4) N,NAV,IP
READ (5,3). (BU(I),I=1,N)
READ (5,3) (BL(I),I=1,N)
READ (5,3) (XS(I),I=1,N)
READ (5,4). NT,NPR
WRITE (6,5) N

      DO 1 I=1,N
1 WRITE (6,6) I,BU(I),I,BL(I),I,XS(I)

      WRITE (6,7) NT
      WRITE (6,8)
      CALL BOXPLX (N,NAV,NPR,NT,2.0,XS,IP,BU,BL,OBJ,IER)
      WRITE (6,9)

      DO 2 I=1,N
2 WRITE (6,10) I,XS(I)

      WRITE (6,11) OBJ
      WRITE (6,12) IER
      STOP

      FORMAT (8F10.5)
3 FORMAT (3I10)
4 FORMAT (.1,.1) VARIABLES - UPPER LIMIT(BU), LOWER LIMIT(BL), START
5 FORMAT (VALUE(XS)) - NUMBER OF VARIABLES=.15,/,/
6 FORMAT (.1, BU(.12,.1)=.F10.5,.12,.1)=.F10.5,.12,.1)=.F10.5, XS(.1,MNBP
7 FORMAT (/, NUMBER OF TRIALS TO BE ALLOWED=.1,I10)
8 FORMAT (.1,)
9 FORMAT (.1,.1) OUTPUT RESULTS,/,/, COORDINATES OF MINIMUM,/)
10 FORMAT (/, Z(.12,.1)=.1PIE15.8)
11 FORMAT (/, TOTAL COST IS =.1PIE15.8)
12 FORMAT (/, ERROR CODE IS =.12)
END

```


SUBROUTINE RBMEAS

This subrcutine measures the range and bearing of the forward and after stations which is required of subrcutine BDGRAS. This is done with trigncmetric functions as shown in chapter II. The subroutine is specifically designed to circumvent any ambiguities usually associated with these functions.

It is the basis of the decoupling of the two RAS ships that this thesis is based.

SUBROUTINE RBMEAS

```

SUBROUTINE RBMEAS (N, PSIA, X1, Y1, PSIB, X2, Y2, D, R1, B1, BB1, R2, B2, BB2)
SUBROUTINE TO CALCULATE THE RELATIVE POSITIONS OF SHIP A (SUPPLY)
TO SHIP B (RECEIVING)
PSIA - HEADING OF SHIP A (RAD)
PSIB - HEADING OF SHIP B (RAD)
X1, Y1 - COORDINATES OF SHIP A
X2, Y2 - COORDINATES OF SHIP B
D - DISTANCE BETWEEN SENSORS (ALSO BETWEEN REFLECTORS) (SAME ON
      BOTH SHIPS)
R1, R2 - FWD, AFT MEASURED DISTANCE
B1, B2 - FWD, AFT RELATIVE BEARINGS (RAD)
BB1, BB2 - FWD, AFT RELATIVE BEARINGS (RAD) WITH SIGN
IF N SET TO 1, RBMEAS SUBROUTINE WILL BE USED
IF (N.EQ.1) GO TO 7
ANGLES AND DISTANCES FOR SENSOR INPUT
SDF1 = (D/2.0)*SIN(PSIA)
SDF2 = (D/2.0)*SIN(PSIB)
SDF3 = (D/2.0)*COS(PSIA)
SDF4 = (D/2.0)*COS(PSIB)
FWD DISTANCE
ADFX = SDF3+X1-SDF4-X2
ADFY = SDF1+Y1-SDF2-Y2
R1 = SQT(ADFX**2+ADFY**2)
AFT DISTANCE
ADAX = -SDF3+X1+SDF4-X2
ADAY = -SDF1+Y1+SDF2-Y2
R2 = SQT(ADAX**2+ADAY**2)
FWD ANGLE
IF (ADFY.EQ.0.0) GO TO 2
BB1 = ATAN2(ADFY, ADFX)
1 BB1 = BB1-PSIB
GO TO 3
2 BB1 = 1.570796327
IF (ADFY.LE.0.0) BB1=-BB1
GO TO 1
AFT ANGLE
IF (ADAX.EQ.0.0) GO TO 5
BB2 = ATAN2(ADAX, ADAY)
4 BB2 = BB2-PSIB
GO TO 6
5 BB2 = 1.570796327

```

RBMS 10
RBMS 20
RBMS 30
RBMS 40
RBMS 50
RBMS 60
RBMS 70
RBMS 80
RBMS 90
RBMS 100
RBMS 110
RBMS 120
RBMS 130
RBMS 140
RBMS 150
RBMS 160
RBMS 170
RBMS 180
RBMS 190
RBMS 200
RBMS 210
RBMS 220
RBMS 230
RBMS 240
RBMS 250
RBMS 260
RBMS 270
RBMS 280
RBMS 290
RBMS 300
RBMS 310
RBMS 320
RBMS 330
RBMS 340
RBMS 350
RBMS 360
RBMS 370
RBMS 380
RBMS 390
RBMS 400
RBMS 410
RBMS 420
RBMS 430



IF (ADAY.LE.0.0) BB2=-BB2
GO TO 4
6 CONTINUE
7 RETURN
END

RBMS 440
RBMS 450
RBMS 460
RBMS 470
RBMS 480

FUNCTION RKLDEQ

This function is the Runge-Kutta-Gill forth-order integration used in all optimization runs. It is programmed locally and is part of the IBM 360 SSP library. A full explanation and description is shown in the first few pages of the function listing.

FUNCTION RKLDEQ

FUNCTION RKLDEQ (FORTRAN 4,G/H) OR ASSEMBLER LANGUAGE)
IDENTIFICATION D2-NPS-RKLDEQ, CHECKED OUT BY R. HILLEARY, 4/67.

PURPOSE
THIS ROUTINE SOLVES A SYSTEM OF N FIRST-ORDER ORDINARY DIFFERENTIAL EQUATIONS BY THE RUNGE-KUTTA-GILL FOURTH-ORDER METHOD. ALL CALCULATIONS ARE IN DOUBLE-PRECISION.

USAGE --(WHEN USED BY FORTRAN CALLING PROGRAM)

S = RKLDEQ (N,Y,F,X,H,NT)

FOUR ENTRIES ARE REQUIRED TO ADVANCE THE SOLUTION FROM X TO X + H WHERE H IS THE INCREMENT. SEE SAMPLE PROBLEM FOR MORE INFORMATION

DESCRIPTION OF PARAMETERS

- N - NUMBER OF FIRST-ORDER EQUATIONS IN SYSTEM TO BE SOLVED. (0.LE.N.LE.25).
- Y - NAME OF LINEAR ARRAY OF LENGTH AT LEAST N, IN WHICH SOLUTION VALUES WILL BE STORED BY RKLDEQ. THE CALLING PROGRAM SHOULD SUPPLY INITIAL VALUES BEFORE FIRST ENTRY.
- F - NAME OF LINEAR ARRAY OF LENGTH AT LEAST N, IN WHICH THE DERIVATIVES, COMPUTED IN USER'S CALLING PROGRAM, ARE STORED.
- X - THE INDEPENDENT VARIABLE, WHICH IS ADVANCED WITHIN RKLDEQ.
- H - THE INCREMENT FOR X, WHICH MAY BE CHANGED AT THE END OF ANY INTERVAL. (WHEN S=2.0)
- NT - AN INTEGER WHICH COUNTS THE NUMBER OF TIMES ENTRY TO RKLDEQ HAS BEEN MADE DURING THE CURRENT INTERVAL. IT MUST BE INITIALLY SET TO ZERO BY USER BEFORE FIRST CALL OF RKLDEQ. SUBSEQUENTLY IT SHOULD NOT BE CHANGED BY USER.
- S - A SWITCH TO BE TESTED BY USER UPON RETURN FROM RKLDEQ. IF S = 1.0, THE CALLING PROGRAM SHOULD NOW COMPUTE VALUES OF F, USING CURRENT VALUES OF X AND Y, AND THEN RETURN TO RKLDEQ. IF S=2.0, AND END OF PRESENT INTERVAL HAS BEEN REACHED. USER SHOULD STORE AND/OR OUTPUT CURRENT X AND/OR Y AND TEST FOR END OF COMPUTATION.

SEE SAMPLE PROBLEM.

CC

RKLDF003
RKLDF004
RKLDF005
RKLDF006
RKLDF007
RKLDF008
RKLDF009
RKLDF010
RKLDF011
RKLDF012
RKLDF013
RKLDF014
RKLDF015
RKLDF016
RKLDF017
RKLDF018
RKLDF019
RKLDF020
RKLDF021
RKLDF022
RKLDF023
RKLDF024
RKLDF025
RKLDF026
RKLDF027
RKLDF028
RKLDF029
RKLDF030
RKLDF031
RKLDF032
RKLDF033
RKLDF034
RKLDF035
RKLDF036
RKLDF037
RKLDF038
RKLDF039
RKLDF040
RKLDF041
RKLDF042
RKLDF043
RKLDF044


```

REMARKS
Y,F,X,AND H ARE DOUBLE PRECISION (REAL*8), RKLDEQ IS REAL*4.
MAXIMUM N IS NOW 25.
RKLDFF045
RKLDFF046
RKLDFF047
RKLDFF048
RKLDFF049
RKLDFF050
RKLDFF051
RKLDFF052
RKLDFF053
RKLDFF054
RKLDFF055
RKLDFF056
RKLDFF057
RKLDFF058
RKLDFF059
RKLDFF060
RKLDFF061
RKLD06A
RKLDFF062
RKLDFF063
RKLDFF064
RKLDFF065
RKLDFF066
RKLDFF067
RKLDFF068
RKLDFF069
RKLDFF070
RKLDFF071
RKLDFF072
RKLDFF073
RKLDFF074
RKLDFF075
RKLDFF076
RKLDFF077
RKLDFF078
RKLDFF079
RKLDFF080
RKLDFF081
RKLDFF082
RKLDFF083
RKLDFF084
RKLDFF085
RKLDFF086
RKLDFF087
RKLDFF088
RKLDFF089
RKLDFF090
RKLDFF091

NOTE
TWO DECKS EXIST FOR THIS FUNCTION. ONE IS IN F4 SOURCE LANGUAGE
THE OTHER IS ASSEMBLER LANGUAGE AND IS TO BE CALLED WITH A
FORTRAN TYPE CALLING SEQUENCE. RKLDEQ IS RETURNED IN F.P.REG.0

SAMPLE PROBLEM
DIMENSION Y(2), F(2)
REAL*8 Y,F,X,H
Y(1)=0.D0
Y(2)=1.D0
X=1.D0
H=.01D0
NT=0
WRITE (6,11) X,Y(1),Y(2)
CALCULATION OF DERIVATIVES0
1 F(1)=X*Y(1)-Y(2)**2
F(2)=Y(1)+DSQRT(X)
S=RKLDEQ(2,Y,F,X,H,NT)
IF (S-1.0) 10,1,2
ERROR STOP0
10 STOP
2 WRITE (6,11) X,Y(1),Y(2)
11 FORMAT (1X,3D25.13)
TEST FOR END OF COMPUTATION0
IF (DABS(X-2.5D0)-1.D-5) 3,3,1
3 STOP
END
.....
FORTRAN 4 VERSION OF RUNGE-KUTTA-GILL ROUTINE
X,Y,F,H ARE DOUBLE-PRECISION. MAX N = 25
FUNCTION RKLDEQ (N,Y,F,X,H,NT)
REAL*8 Y,F,X,H,Q,H1,H2,H3,H6
DIMENSION Y(1),F(1),Q(25)
NT=NT+1
GO TO (1,2,3,4),NT
1 H1=H

```


RKLDF092
 RKLDF093
 RKLDF094
 RKLDF095
 RKLDF096
 RKLDF097
 RKLDF098
 RKLDF099
 RKLDF100
 RKLDF101
 RKLDF102
 RKLDF103
 RKLDF104
 RKLDF105
 RKLDF106
 RKLDF107
 RKLDF108
 RKLDF109
 RKLDF110
 RKLDF111
 RKLDF112
 RKLDF113
 RKLDF114
 RKLDF115
 RKLDF116
 RKLDF117
 RKLDF118
 RKLDF119
 RKLDF120

```

H2 = H1 * .5D0
H3 = H1 * 2.D0
H6 = H1/6.D0
DO 11 J = 1,N
  11 Q(J) = 0.D0
  A = .5D0
  X = X + H2
  GO TO 5
C
2 A = .2928932188134525
  GO TO 5
C
3 A = 1.7071067811865475
  X = X + H2
  GO TO 5
C
4 DO 41 I = 1,N
  41 Y(I) = Y(I) + H6 * F(I) - Q(I)/3.D0
  NT = 0
  RKLDEQ = 2.
  GO TO 6
C
5 DO 51 L = 1,N
  51 Y(L) = Y(L) + A * (H * F(L) - Q(L))
  Q(L) = H3 * A * F(L) + (1.D0 - 3.D0 * A) * Q(L)
  RKLDEQ = 1.
C
6 RETURN
END

```


SUBROUTINE SLOPES

This subroutine contains the table look-up and interpolation scheme for the interactive forces and moments presented in the RAS environment. It is long and must be pre-compiled for most of the DSL simulation programs shown in this thesis.

SUBROUTINE SLOPES (DX,DY,YY1,YY2,YN1,YN2)		TABLE LOOK-UP AND INTERPOLATION	
DIMENSION Z(23,16), W(23,16), X(23), Y(16)			
X(1)	= -1.1		
X(2)	= -1.9		
X(3)	= -.8		
X(4)	= -.7		
X(5)	= -.6		
X(6)	= -.5		
X(7)	= -.4		
X(8)	= -.3		
X(9)	= -.2		
X(10)	= -.1		
X(11)	= 0.		
X(12)	= .1		
X(13)	= .2		
X(14)	= .3		
X(15)	= .4		
X(16)	= .5		
X(17)	= .6		
X(18)	= .7		
X(19)	= .8		
X(20)	= .9		
X(21)	= 1.		
X(22)	= 1.		
X(23)	= 1.		
Y(1)	= 0.	10	
Y(2)	= 0.	12	
Y(3)	= 0.	14	
Y(4)	= 0.	16	
Y(5)	= 0.	18	
Y(6)	= 0.	22	
Y(7)	= 0.	24	
Y(8)	= 0.	26	
Y(9)	= 0.	28	
Y(10)	= 0.	30	
Y(11)	= 0.	32	
Y(12)	= 0.	34	
Y(13)	= 0.	36	
Y(14)	= 0.	38	
Y(15)	= 0.	40	
Y(16)	= 0.	42	
Z(1,1)	= 0.	44	
		46	
		48	
		50	
		52	
		54	
		56	
		58	
		60	
		62	
		64	
		66	
		68	
		70	
		72	
		74	
		76	
		78	
		80	
		82	
		84	
		86	
		88	
		90	
		92	
		94	
		96	
		98	
		100	
		102	
		104	
		106	
		108	
		110	
		112	
		114	
		116	
		118	
		120	
		122	
		124	
		126	
		128	
		130	
		132	
		134	
		136	
		138	
		140	
		142	
		144	
		146	
		148	
		150	
		152	
		154	
		156	
		158	
		160	
		162	
		164	
		166	
		168	
		170	
		172	
		174	
		176	
		178	
		180	
		182	
		184	
		186	
		188	
		190	
		192	
		194	
		196	
		198	
		200	
		202	
		204	
		206	
		208	
		210	
		212	
		214	
		216	
		218	
		220	
		222	
		224	
		226	
		228	
		230	
		232	
		234	
		236	
		238	
		240	
		242	
		244	
		246	
		248	
		250	
		252	
		254	
		256	
		258	
		260	
		262	
		264	
		266	
		268	
		270	

253

Z(4,2) 33.
Z(4,3) -28.
Z(4,4) -23.
Z(4,5) -18.
Z(4,6) -16.
Z(4,7) -14.
Z(4,8) -12.
Z(4,9) -10.
Z(4,10) -8.
Z(4,11) -6.
Z(4,12) -4.
Z(4,13) -2.
Z(4,14) 0.
Z(4,15) 0.
Z(4,16) 0.
Z(5,1) -38.
Z(5,2) -31.
Z(5,3) -26.
Z(5,4) -21.
Z(5,5) -18.
Z(5,6) -14.
Z(5,7) -12.
Z(5,8) 0.
Z(5,9) -8.
Z(5,10) -6.
Z(5,11) -4.
Z(5,12) -2.
Z(5,13) 0.
Z(5,14) 0.
Z(5,15) 0.
Z(5,16) 0.
Z(6,1) -28.
Z(6,2) -24.
Z(6,3) -20.
Z(6,4) -15.
Z(6,5) -12.
Z(6,6) -10.
Z(6,7) -8.
Z(6,8) -6.
Z(6,9) -4.
Z(6,10) -2.
Z(6,11) 0.
Z(6,12) 0.
Z(6,13) 0.
Z(6,14) 0.
Z(6,15) 0.
Z(7,1) -15.

SL0P 920
SL0P 930
SL0P 940
SL0P 950
SL0P 960
SL0P 970
SL0P 980
SL0P 990
SL0P 1000
SL0P 1010
SL0P 1020
SL0P 1030
SL0P 1040
SL0P 1050
SL0P 1060
SL0P 1070
SL0P 1080
SL0P 1090
SL0P 1100
SL0P 1110
SL0P 1120
SL0P 1130
SL0P 1140
SL0P 1150
SL0P 1160
SL0P 1170
SL0P 1180
SL0P 1190
SL0P 1200
SL0P 1210
SL0P 1220
SL0P 1230
SL0P 1240
SL0P 1250
SL0P 1260
SL0P 1270
SL0P 1280
SL0P 1290
SL0P 1300
SL0P 1310
SL0P 1320
SL0P 1330
SL0P 1340
SL0P 1350
SL0P 1360
SL0P 1370
SL0P 1380
SL0P 1390

z(7,2)
z(7,3)
z(7,4)
z(7,5)
z(7,6)
z(7,7)
z(7,8)
z(7,9)
z(7,10)
z(7,11)
z(7,12)
z(7,13)
z(7,14)
z(7,15)
z(7,16)
z(8,1)
z(8,2)
z(8,3)
z(8,4)
z(8,5)
z(8,6)
z(8,7)
z(8,8)
z(8,9)
z(8,10)
z(8,11)
z(8,12)
z(8,13)
z(8,14)
z(8,15)
z(8,16)
z(9,1)
z(9,2)
z(9,3)
z(9,4)
z(9,5)
z(9,6)
z(9,7)
z(9,8)
z(9,9)
z(9,10)
z(9,11)
z(9,12)
z(9,13)
z(9,14)
z(9,15)
z(9,16)
z(10,1)
-11.
-9.
-6.
-4.
-2.
0.
0.
0.
0.
0.
0.
0.
0.
0.
0.
4.
6.
7.
8.
9.
10.
7.
6.
4.
2.
0.
0.
0.
0.
0.
0.
0.
27.
25.
22.
21.
20.
18.
16.
14.
12.
10.
8.
6.
4.
2.
0.
52.

slop1400
slop1410
slop1420
slop1430
slop1440
slop1450
slop1460
slop1470
slop1480
slop1490
slop1500
slop1510
slop1520
slop1530
slop1540
slop1550
slop1560
slop1570
slop1580
slop1590
slop1600
slop1610
slop1620
slop1630
slop1640
slop1650
slop1660
slop1670
slop1680
slop1690
slop1700
slop1710
slop1720
slop1730
slop1740
slop1750
slop1760
slop1770
slop1780
slop1790
slop1800
slop1810
slop1820
slop1830
slop1840
slop1850
slop1860
slop1870

Z(13,2)78.
Z(13,3)69.
Z(13,4)60.
Z(13,5)53.
Z(13,6)48.
Z(13,7)43.
Z(13,8)38.
Z(13,9)34.
Z(13,10)31.
Z(13,11)28.
Z(13,12)25.
Z(13,13)22.
Z(13,14)19.
Z(13,15)17.
Z(13,16)15.
Z(14,1)80.
Z(14,2)70.
Z(14,3)63.
Z(14,4)55.
Z(14,5)50.
Z(14,6)45.
Z(14,7)40.
Z(14,8)36.
Z(14,9)33.
Z(14,10)30.
Z(14,11)29.
Z(14,12)26.
Z(14,13)23.
Z(14,14)20.
Z(14,15)18.
Z(14,16)16.
Z(15,1)64.
Z(15,2)56.
Z(15,3)51.
Z(15,4)46.
Z(15,5)41.
Z(15,6)38.
Z(15,7)36.
Z(15,8)34.
Z(15,9)32.
Z(15,10)30.
Z(15,11)28.
Z(15,12)26.
Z(15,13)24.
Z(15,14)22.
Z(15,15)20.
Z(15,16)18.
Z(16,1)45.

SLOP2360
SLOP2370
SLOP2380
SLOP2390
SLOP2400
SLOP2410
SLOP2420
SLOP2430
SLOP2440
SLOP2450
SLOP2460
SLOP2470
SLOP2480
SLOP2490
SLOP2500
SLOP2510
SLOP2520
SLOP2530
SLOP2540
SLOP2550
SLOP2560
SLOP2570
SLOP2580
SLOP2590
SLOP2600
SLOP2610
SLOP2620
SLOP2630
SLOP2640
SLOP2650
SLOP2660
SLOP2670
SLOP2680
SLOP2690
SLOP2700
SLOP2710
SLOP2720
SLOP2730
SLOP2740
SLOP2750
SLOP2760
SLOP2770
SLOP2780
SLOP2790
SLOP2800
SLOP2810
SLOP2820
SLOP2830

[illegible]

W(2,3)
W(2,4)
W(2,5)
W(2,6)
W(2,7)
W(2,8)
W(2,9)
W(2,10)
W(2,11)
W(2,12)
W(2,13)
W(2,14)
W(2,15)
W(2,16)
W(3,1)
W(3,2)
W(3,3)
W(3,4)
W(3,5)
W(3,6)
W(3,7)
W(3,8)
W(3,9)
W(3,10)
W(3,11)
W(3,12)
W(3,13)
W(3,14)
W(3,15)
W(3,16)
W(4,1)
W(4,2)
W(4,3)
W(4,4)
W(4,5)
W(4,6)
W(4,7)
W(4,8)
W(4,9)
W(4,10)
W(4,11)
W(4,12)
W(4,13)
W(4,14)
W(4,15)
W(4,16)
W(5,1)

= 10.
= 19.
= 8.
= 6.
= 4.
= 2.
= 0.
= 0.
= 0.
= 0.
= 0.
= 0.
= 0.
= 0.
= 14.
= 14.
= 11.
= 10.
= 9.
= 8.
= 7.
= 6.
= 5.
= 4.
= 3.
= 2.
= 1.
= 0.
= 0.
= 0.
= 14.
= 12.
= 10.
= 8.
= 7.
= 6.
= 5.
= 4.
= 3.
= 2.
= 1.
= 0.
= 0.
= 0.
= 7.

SLOP4280
SLOP4290
SLOP4300
SLOP4310
SLOP4320
SLOP4330
SLOP4340
SLOP4350
SLOP4360
SLOP4370
SLOP4380
SLOP4390
SLOP4400
SLOP4410
SLOP4420
SLOP4430
SLOP4440
SLOP4450
SLOP4460
SLOP4470
SLOP4480
SLOP4490
SLOP4500
SLOP4510
SLOP4520
SLOP4530
SLOP4540
SLOP4550
SLOP4560
SLOP4570
SLOP4580
SLOP4590
SLOP4600
SLOP4610
SLOP4620
SLOP4630
SLOP4640
SLOP4650
SLOP4660
SLOP4670
SLOP4680
SLOP4690
SLOP4700
SLOP4710
SLOP4720
SLOP4730
SLOP4740
SLOP4750

W(5,2)	6.	=
W(5,3)	5.	=
W(5,4)	4.	=
W(5,5)	3.	=
W(5,6)	2.	=
W(5,7)	1.	=
W(5,8)	0.	=
W(5,9)	0.	=
W(5,10)	0.	=
W(5,11)	0.	=
W(5,12)	0.	=
W(5,13)	0.	=
W(5,14)	0.	=
W(5,15)	0.	=
W(5,16)	0.	=
W(6,1)	-5.	=
W(6,2)	-5.	=
W(6,3)	-4.	=
W(6,4)	-4.	=
W(6,5)	-4.	=
W(6,6)	-4.	=
W(6,7)	-3.	=
W(6,8)	-3.	=
W(6,9)	-3.	=
W(6,10)	-2.	=
W(6,11)	-2.	=
W(6,12)	-1.	=
W(6,13)	-1.	=
W(6,14)	0.	=
W(6,15)	0.	=
W(6,16)	0.	=
W(7,1)	-18.	=
W(7,2)	-16.	=
W(7,3)	-14.	=
W(7,4)	-12.	=
W(7,5)	-9.	=
W(7,6)	-8.	=
W(7,7)	-7.	=
W(7,8)	-6.	=
W(7,9)	-5.	=
W(7,10)	-4.	=
W(7,11)	-3.	=
W(7,12)	-2.	=
W(7,13)	-1.	=
W(7,14)	0.	=
W(7,15)	0.	=
W(7,16)	0.	=
W(8,1)	-30.	=

S(4,760)	LOP4760
S(4,770)	LOP4770
S(4,780)	LOP4780
S(4,790)	LOP4790
S(4,800)	LOP4800
S(4,810)	LOP4810
S(4,820)	LOP4820
S(4,830)	LOP4830
S(4,840)	LOP4840
S(4,850)	LOP4850
S(4,860)	LOP4860
S(4,870)	LOP4870
S(4,880)	LOP4880
S(4,890)	LOP4890
S(4,900)	LOP4900
S(4,910)	LOP4910
S(4,920)	LOP4920
S(4,930)	LOP4930
S(4,940)	LOP4940
S(4,950)	LOP4950
S(4,960)	LOP4960
S(4,970)	LOP4970
S(4,980)	LOP4980
S(4,990)	LOP4990
S(5,000)	LOP5000
S(5,010)	LOP5010
S(5,020)	LOP5020
S(5,030)	LOP5030
S(5,040)	LOP5040
S(5,050)	LOP5050
S(5,060)	LOP5060
S(5,070)	LOP5070
S(5,080)	LOP5080
S(5,090)	LOP5090
S(5,100)	LOP5100
S(5,110)	LOP5110
S(5,120)	LOP5120
S(5,130)	LOP5130
S(5,140)	LOP5140
S(5,150)	LOP5150
S(5,160)	LOP5160
S(5,170)	LOP5170
S(5,180)	LOP5180
S(5,190)	LOP5190
S(5,200)	LOP5200
S(5,210)	LOP5210
S(5,220)	LOP5220
S(5,230)	LOP5230

W(8,2)	26.
W(8,3)	-22.
W(8,4)	-19.
W(8,5)	-17.
W(8,6)	-15.
W(8,7)	-13.
W(8,8)	-11.
W(8,9)	-9.
W(8,10)	-7.
W(8,11)	-5.
W(8,12)	-3.
W(8,13)	-1.
W(8,14)	0.
W(8,15)	0.
W(8,16)	0.
W(9,1)	-4.
W(9,2)	-3.
W(9,3)	-3.
W(9,4)	-2.
W(9,5)	-2.
W(9,6)	-1.
W(9,7)	-1.
W(9,8)	-1.
W(9,9)	-1.
W(9,10)	-1.
W(9,11)	-1.
W(9,12)	-1.
W(9,13)	-1.
W(9,14)	-1.
W(9,15)	-1.
W(9,16)	-1.
W(10,1)	4.
W(10,2)	-3.
W(10,3)	-3.
W(10,4)	-2.
W(10,5)	-2.
W(10,6)	-2.
W(10,7)	-1.
W(10,8)	-1.
W(10,9)	-1.
W(10,10)	-1.
W(10,11)	-1.
W(10,12)	-1.
W(10,13)	-1.
W(10,14)	-1.
W(10,15)	-1.
W(10,16)	-1.
W(11,1)	-4.

S	OP	52	40
S	OP	52	50
S	OP	52	60
S	OP	52	70
S	OP	52	80
S	OP	52	90
S	OP	53	00
S	OP	53	10
S	OP	53	20
S	OP	53	30
S	OP	53	40
S	OP	53	50
S	OP	53	60
S	OP	53	70
S	OP	53	80
S	OP	53	90
S	OP	54	00
S	OP	54	10
S	OP	54	20
S	OP	54	30
S	OP	54	40
S	OP	54	50
S	OP	54	60
S	OP	54	70
S	OP	54	80
S	OP	54	90
S	OP	55	00
S	OP	55	10
S	OP	55	20
S	OP	55	30
S	OP	55	40
S	OP	55	50
S	OP	55	60
S	OP	55	70
S	OP	55	80
S	OP	55	90
S	OP	56	00
S	OP	56	10
S	OP	56	20
S	OP	56	30
S	OP	56	40
S	OP	56	50
S	OP	56	60
S	OP	56	70
S	OP	56	80
S	OP	56	90
S	OP	57	00
S	OP	57	10

W(11,1,1) 0.37
W(11,1,2) -0.33
W(11,1,3) -0.28
W(11,1,4) -0.25
W(11,1,5) -0.20
W(11,1,6) -0.17
W(11,1,7) -0.14
W(11,1,8) -0.11
W(11,1,9) -0.09
W(11,1,10) -0.07
W(11,1,11) -0.05
W(11,1,12) -0.03
W(11,1,13) -0.02
W(11,1,14) -0.01
W(11,1,15) 0.01
W(11,1,16) 0.03
W(12,2,1) 0.37
W(12,2,2) -0.33
W(12,2,3) -0.30
W(12,2,4) -0.26
W(12,2,5) -0.23
W(12,2,6) -0.20
W(12,2,7) -0.17
W(12,2,8) -0.14
W(12,2,9) -0.11
W(12,2,10) -0.09
W(12,2,11) -0.07
W(12,2,12) -0.05
W(12,2,13) -0.03
W(12,2,14) -0.02
W(12,2,15) -0.01
W(12,2,16) 0.01
W(13,3,1) 0.29
W(13,3,2) -0.26
W(13,3,3) -0.25
W(13,3,4) -0.22
W(13,3,5) -0.19
W(13,3,6) -0.17
W(13,3,7) -0.15
W(13,3,8) -0.13
W(13,3,9) -0.11
W(13,3,10) -0.09
W(13,3,11) -0.07
W(13,3,12) -0.05
W(13,3,13) -0.03
W(13,3,14) -0.02
W(13,3,15) -0.01
W(13,3,16) 0.01
W(14,4,1) -0.18

SLOP5720
SLOP5730
SLOP5740
SLOP5750
SLOP5760
SLOP5770
SLOP5780
SLOP5790
SLOP5800
SLOP5810
SLOP5820
SLOP5830
SLOP5840
SLOP5850
SLOP5860
SLOP5870
SLOP5880
SLOP5890
SLOP5900
SLOP5910
SLOP5920
SLOP5930
SLOP5940
SLOP5950
SLOP5960
SLOP5970
SLOP5980
SLOP5990
SLOP6000
SLOP6010
SLOP6020
SLOP6030
SLOP6040
SLOP6050
SLOP6060
SLOP6070
SLOP6080
SLOP6090
SLOP6100
SLOP6110
SLOP6120
SLOP6130
SLOP6140
SLOP6150
SLOP6160
SLOP6170
SLOP6180
SLOP6190

W(14,2) = 16.
 W(14,3) = -15.
 W(14,4) = -13.
 W(14,5) = -12.
 W(14,6) = -10.
 W(14,7) = -9.
 W(14,8) = -8.
 W(14,9) = -7.
 W(14,10) = -6.
 W(14,11) = -5.
 W(14,12) = -4.
 W(14,13) = -3.
 W(14,14) = -2.
 W(14,15) = -1.
 W(14,16) = 0.
 W(15,1) = 5.
 W(15,2) = -4.
 W(15,3) = -4.
 W(15,4) = -4.
 W(15,5) = -4.
 W(15,6) = -4.
 W(15,7) = -3.
 W(15,8) = -3.
 W(15,9) = -3.
 W(15,10) = 3.
 W(15,11) = -2.
 W(15,12) = -2.
 W(15,13) = -1.
 W(15,14) = -1.
 W(15,15) = 0.
 W(15,16) = 0.
 W(16,1) = -4.
 W(16,2) = -4.
 W(16,3) = -3.
 W(16,4) = -3.
 W(16,5) = -2.
 W(16,6) = -2.
 W(16,7) = -2.
 W(16,8) = -1.
 W(16,9) = -1.
 W(16,10) = -1.
 W(16,11) = 0.
 W(16,12) = 0.
 W(16,13) = 0.
 W(16,14) = 0.
 W(16,15) = 0.
 W(16,16) = 0.
 W(17,1) = 11.

SLOP6200
 SLOP6210
 SLOP6220
 SLOP6230
 SLOP6240
 SLOP6250
 SLOP6260
 SLOP6270
 SLOP6280
 SLOP6290
 SLOP6300
 SLOP6310
 SLOP6320
 SLOP6330
 SLOP6340
 SLOP6350
 SLOP6360
 SLOP6370
 SLOP6380
 SLOP6390
 SLOP6400
 SLOP6410
 SLOP6420
 SLOP6430
 SLOP6440
 SLOP6450
 SLOP6460
 SLOP6470
 SLOP6480
 SLOP6490
 SLOP6500
 SLOP6510
 SLOP6520
 SLOP6530
 SLOP6540
 SLOP6550
 SLOP6560
 SLOP6570
 SLOP6580
 SLOP6590
 SLOP6600
 SLOP6610
 SLOP6620
 SLOP6630
 SLOP6640
 SLOP6650
 SLOP6660
 SLOP6670

SLOP7640
 SLOP7650
 SLOP7660
 SLOP7670
 SLOP7680
 SLOP7690
 SLOP7700
 SLOP7710
 SLOP7720
 SLOP7730
 SLOP7740
 SLOP7750
 SLOP7760
 SLOP7770
 SLOP7780
 SLOP7790
 SLOP7800
 SLOP7810
 SLOP7820
 SLOP7830
 SLOP7840
 SLOP7850
 SLOP7860
 SLOP7870
 SLOP7880
 SLOP7890
 SLOP7900
 SLOP7910
 SLOP7920
 SLOP7930
 SLOP7940
 SLOP7950
 SLOP7960
 SLOP7970
 SLOP7980
 SLOP7990
 SLOP8000
 SLOP8010
 SLOP8020
 SLOP8030
 SLOP8040
 SLOP8050
 SLOP8060
 SLOP8070
 SLOP8080
 SLOP8090
 SLOP8100
 SLOP8110

```

W(23,2) = 0.
W(23,3) = 0.
W(23,4) = 0.
W(23,5) = 0.
W(23,6) = 0.
W(23,7) = 0.
W(23,8) = 0.
W(23,9) = 0.
W(23,10) = 0.
W(23,11) = 0.
W(23,12) = 0.
W(23,13) = 0.
W(23,14) = 0.
W(23,15) = 0.
W(23,16) = 0.
IF (ABS(DY).GE..4734) GO TO 1
IF (ABS(DX).GT.1.15) GO TO 1
I = IFIX((DX+1.1002)/.1)+1.
J = (J.LT.1) J = 1
IF (I.LT.1) I = 1
IF (I.GT.23) I = 23
IF (J.GT.16) J = 16
K = 24-I
DELX = DX-X(I)
DELY = ABS(DY)-Y(J)
DELK = DX-X(K)
IF ((I.EQ.23) OR (J.EQ.16)) GO TO 2
DYD1 = DELX*(Z(I+1,J))-Z(I,J))+DELY*(Z(I,J+1))-Z(I,J))
DYD2 = DELK*(Z(K+1,J))-Z(K,J))+DELY*(Z(K,J+1))-Z(K,J))
DYND1 = DELX*(W(I+1,J))-W(I,J))+DELY*(W(I,J+1))-W(I,J))
DYND2 = DELK*(W(K+1,J))-W(K,J))+DELY*(W(K,J+1))-W(K,J))
YY1 = (Z(I,J)+DYD1)*1.E-05
YY2 = -(Z(K,J)+DYD2)*1.E-05
YN1 = (W(I,J)+DYND1)*1.E-05
YN2 = -(W(K,J)+DYND2)*1.E-05
IF (DY.LT.0.) GO TO 3
RETURN
1 YY1 = 0.
  YY2 = 0.
  YN1 = 0.
  YN2 = 0.
RETURN
2 YY1 = Z(I,J)*1.E-05
  YY2 = -Z(K,J)*1.E-05
  YN1 = W(I,J)*1.E-05
  YN2 = -W(K,J)*1.E-05
IF (DY.LT.0.) GO TO 3
  
```


SLOP8120
SLOP8130
SLOP8140
SLOP8150
SLOP8160
SLOP8170
SLOP8180

3
RETURN -YY1
YY1 = -YY2
YY2 = -YN1
YN1 = -YN2
YN2 = -RETURN
RETURN
END

FUNCTION SPINIT

This function was designed to aid in initialization problems associated with the DSL function DELY. The effect is that the function initializes the delay loop until it can be self-supportive.

FUNCTION SPINIT

```
FUNCTION SPINIT (SPDDEL, TIME, SPD0)  
IF (TIME.GT.0.5) GO TO 1  
SPINIT = SPD0  
RETURN  
1 SPINIT = SPDDEL  
RETURN  
END
```

```
SPIN 10  
SPIN 20  
SPIN 30  
SPIN 40  
SPIN 50  
SPIN 60  
SPIN 70
```


FUNCTION SPDCTR

This function is the speed control used during heading control development. It is used directly as the speed of the control ship with information of the speed of the two ships and the longitudinal position ADX. It contains a linear function at ± 1.0 ship lengths to a dead zone of ± 0.001 centered about the alongside position (0.0).

FUNCTION SPDCTR

```
FUNCTION SPDCTR (ADX,U01,U02)
  IF (ADX.LT.-1.0) GO TO 1
  IF (ADX.GT.1.0) GO TO 2
  IF (ABS(ADX).LT.0.001) GO TO 3
  SPDCTR = -ADX*(U02-U01)+U01
  RETURN
1 SPDCTR = U02
2 RETURN = U02-U01
3 RETURN = U01
END
```

SPCR 10
SPCR 20
SPCR 30
SPCR 40
SPCR 50
SPCR 60
SPCR 70
SPCR 80
SPCR 90
SPCR 100
SPCR 110
SPCR 120
SPCR 130

FUNCTION SPDOFC

This function is identical to SPDREC except that the ability to offset the alongside position (0.0) is incorporated. This is the speed control function in its final development form.

FUNCTION SPD0FC

```

FUNCTION SPD0FC (ADX, SPD01, SPD02, SW, XOFS)
SWTCH = -SW*(SPD02-SPD01)
IF ((ADX-XOFS).LT.SWTCH) GO TO 1
IF ((ADX-XOFS).GT.-SWTCH) GO TO 2
IF (ABS(ADX-XOFS).LT.0.001) GO TO 3
SPD0FC = -(ADX-XOFS)*(SPD02-SPD01)+SPD01
RETURN
1 SPD0FC = SPD02
2 SPD0FC = SPD02-SPD01
3 SPD0FC = SPD01
END

```

```

SPD00 10
SPD00 20
SPD00 30
SPD00 40
SPD00 50
SPD00 60
SPD00 70
SPD00 80
SPD00 90
SPD00 100
SPD00 110
SPD00 120
SPD00 130
SPD00 140

```


FUNCTION SPDREC

This function is similar to SPDCTR except that a switching function is incorporated. This is the function used for optimization of the switching function and is used in the velocity loop simulated in the velocity control section of chapter III.

FUNCTION SPDREC

```
FUNCTION SPDREC (ADX,SPD01,SPD02,SW)
SWTCH = -SW*(SPD02-SPD01)
IF (ADX.LT.SWTCH) GO TO 1
IF (ADX.GT.-SWTCH) GO TO 2
IF (ABS(ADX).LT.0.001) GO TO 3
SPDREC = -ADX*(SPD02-SPD01)+SPD01
RETURN
1 SPDREC = SPD02
2 SPDREC = SPD02-SPD01
3 SPDREC = SPD01
END
```

SPDR 10
SPDR 20
SPDR 30
SPDR 40
SPDR 50
SPDR 60
SPDR 70
SPDR 80
SPDR 90
SPDR 100
SPDR 110
SPDR 120
SPDR 130
SPDR 140

FUNCTION SWCL

This function contains the fifth order polynomial curve fit for the optimal switching position of the speed control loop. Its range of values for SPDDIF are 0.1 to 1.0 normalized speed difference between the two ships.

FUNCTION SWCL

```
FUNCTION SWCL (SPD01, SPD02)
SPDDIF = SPD02-SPD01
SWCL = -2.24869*SPDDIF**5+6.93243*SPDDIF**4-8.04233*SPDDIF**3+4.08
1065*SPDDIF**2-0.409977*SPDDIF+0.554
RETURN
END
```

SWCL
SWCL
SWCL
SWCL
SWCL
SWCL

10
20
30
40
50
60

SUBROUTINE SWITCH

This subroutine contains the gains and mechanisms required for the adaptive gain schedule developed in this thesis. It includes the optimal gains obtained from the heading control optimization runs.

SUBROUTINE SWITCH

```

C
SUBROUTINE SWITCH (DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)
SUBROUTINE TO SWITCH RAS GAINS ONCE SHIPS ALONGSIDE
DDC = DD
IF (IS.EQ.0) DDC = -DD
AMDY = DDC+DA*SIN(AA1)
AMDX = DA*COS(AA1)
IF (ABS(AMDX).GT.1.0) N=1
IF (ABS(AMDX).LT.0.5.AND.ABS(AMDY).LT.0.005) GO TO 1
IF (ABS(AMDY).LT.0.05) GO TO 2
N = 1
RSENS = 1.86642
WTSENS = 2.38692
RGN = 23.41847
VFBG = 4.35162
RETURN
1 RSENS = 1.99765
WTSENS = 0.7357
RGN = 49.97757
VFBG = 0.084028
N = N+1
IF (ABS(BDOT2D).GT.2.0.AND.N.LT.150) VFBG=1.0
RETURN
2 RSENS = 4.0
WTSENS = 2.38692
RGN = 23.41847
VFBG = 4.35162
IF (N.GT.150) GO TO 1
N = 1
RETURN
END
SWITH 10
SWITH 20
SWITH 30
SWITH 40
SWITH 50
SWITH 60
SWITH 70
SWITH 80
SWITH 90
SWITH 100
SWITH 110
SWITH 120
SWITH 130
SWITH 140
SWITH 150
SWITH 160
SWITH 170
SWITH 180
SWITH 190
SWITH 200
SWITH 210
SWITH 220
SWITH 230
SWITH 240
SWITH 250
SWITH 260
SWITH 270
SWITH 280
SWITH 290
SWITH 300

```


SUBROUTINE SWTCHF

This subroutine is identical to SWTCH except that the turn phase gain VFEG is relaxed to allow for offset longitudinal position placement. This is the adaptive gain schedule in its final form.

SUBROUTINE SWITCHF

```

SUBROUTINE SWITCHF (DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
SUBROUTINE TO SWITCH RAS GAINS ONCE SHIPS ALONGSIDE
DDC = DD
IF (IS.EQ.0) DDC = -DD
AMDY = DDC+DA*SIN(AA1)
AMDZ = DA*COS(AA1)
IF (ABS(AMDZ-XOFS).GT.1.0) N=1
IF (ABS(AMDY-XOFS).LT.0.5.AND.ABS(AMDY).LT.0.005) GO TO 1
IF (ABS(AMDY).LT.0.05) GO TO 2
N = 1
RSENS = 1.86642
WTSENS = 2.38692
RGN = 23.41847
VFBG = 4.35162
RETURN
1 RSENS = 1.99765
WTSENS = 0.7357
RGN = 49.97757
VFBG = 0.1
IF (ABS(BDOT2D).GT.2.0.AND.N.LE.150) VFBG=1.0
N = N+1
RETURN
2 RSENS = 4.0
WTSENS = 2.38692
RGN = 23.41847
VFBG = 4.35162
IF (N.GT.150) GO TO 1
N = 1
RETURN
END

```

10 SWITCHF
 20 SWITCHF
 30 SWITCHF
 40 SWITCHF
 50 SWITCHF
 60 SWITCHF
 70 SWITCHF
 80 SWITCHF
 90 SWITCHF
 100 SWITCHF
 110 SWITCHF
 120 SWITCHF
 130 SWITCHF
 140 SWITCHF
 150 SWITCHF
 160 SWITCHF
 170 SWITCHF
 180 SWITCHF
 190 SWITCHF
 200 SWITCHF
 210 SWITCHF
 220 SWITCHF
 230 SWITCHF
 240 SWITCHF
 250 SWITCHF
 260 SWITCHF
 270 SWITCHF
 280 SWITCHF
 290 SWITCHF
 300 SWITCHF

SUBROUTINE TRANS

This subroutine takes the lateral and longitudinal geographic displacements and converts them to actual displacements referenced to the control ship's head. This is done to gain a more realistic reference for subroutine FEMEAS, subroutine SLOPES, function SPDCTR, function SPDCFC, and function SPDREC.

SUBROUTINE TRANS

```

SUBROUTINE TRANS (PSIA,DX,DY,ADX,ADY)
  DXY = SQRT(DX**2+DY**2)
  AXY = ARSIN(-DY/DXY)
  IF (DX.LT.0.) AXY = 3.141592654-AXY
  AT = AXY+PSIA
  ADY = -DXY*SIN(AT)
  ADX = +DXY*COS(AT)
  RETURN
END

```

```

10
20
30
40
50
60
70
80
90
TRNS
TRNS
TRNS
TRNS
TRNS
TRNS
TRNS
TRNS

```


FUNCTION XLIMIT

This function was developed to allow the LIMIT function of DSL to be incorporated in the optimization runs. It is a saturation amplifier with a gain of 1.0, and upper limit of UL, and a lower limit of XLL.

FUNCTION XLIMIT

```
FUNCTION XLIMIT (XLL,UL,FUNCT)
  FUNCU = FUNCT
  IF (FUNCT.LT.XLL) FUNCU=XLL
  IF (FUNCT.GT.UL) FUNCU=UL
  XLIMIT = FUNCU
  RETURN
END
```

```
XLMT 10
XLMT 20
XLMT 30
XLMT 40
XLMT 50
XLMT 60
XLMT 70
```


APPENDIX B

The final form of the simulation program, with all its subroutines and functions, is a very complex and complicated maze. To aid in following its progression, this appendix contains a detailed block diagram of the program with each variable listed in its computer variable name. Each page contains a functional part of the simulation with inputs and outputs shown cross referenced to their origin and destination.

The following is a list of the block diagrams contained in this appendix in the order in which they appear:

Ship A (Reference Ship) Simulation

Ship A Heading Simulation

Ship A Speed Simulation

Ship B (Control Ship) Simulation

Ship B Heading Simulation

Ship B Speed Simulation

Subroutine RBMEAS

Range Measurement

Bearing Measurement

Subroutine HDGRAS

Heading Control Loop

Auxiliary Functions

Yaw Conversion

Coordinate Conversion

Feedback Loop

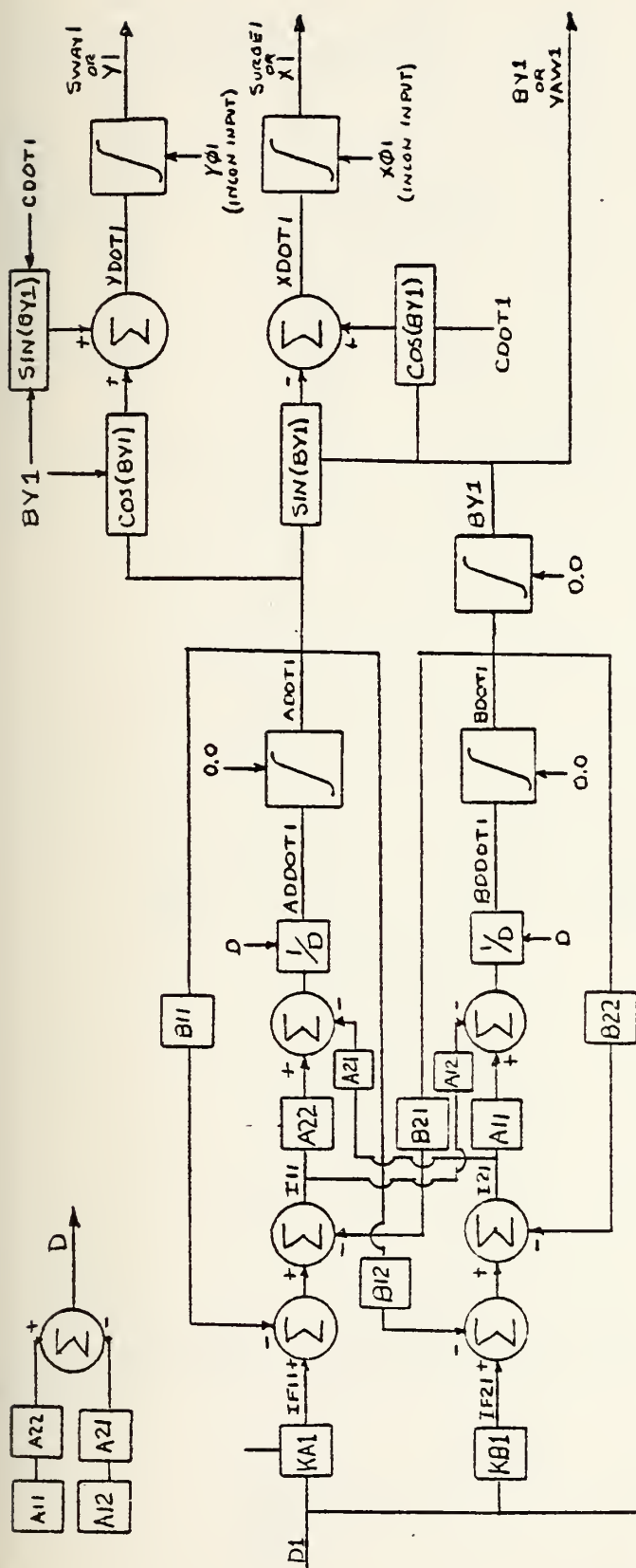
Rudder Modeling

Ship A Rudder

Ship B Rudder

Wave Generator

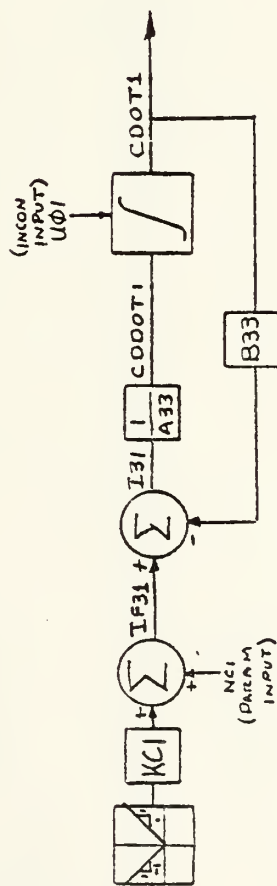
Wave Direction
Random Variable
Wave Encounter
Wave Components



Ship A Heading Simulation

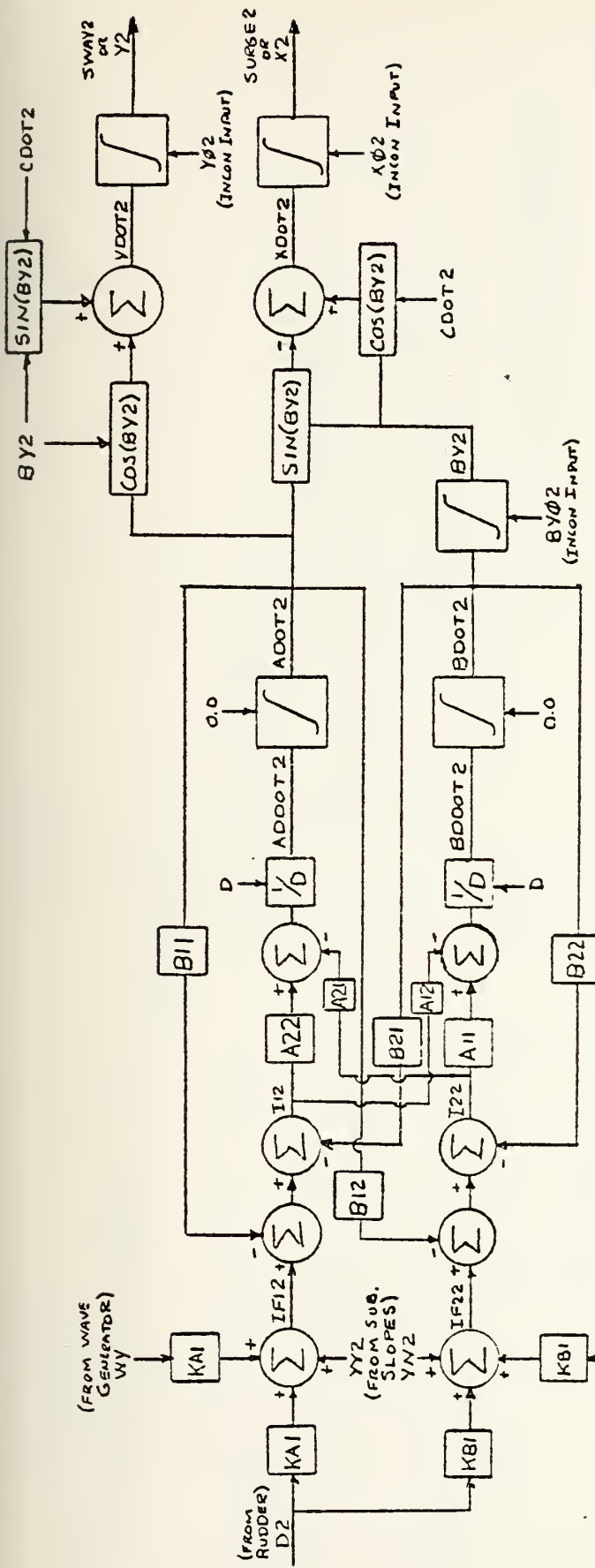
GAIN CROSS REFERENCE

A11 = MYVD
B11 = -YV
A21 = -YR
B21 = MYRD
A12 = -MYD
B12 = -NV
A22 = IZNRD
B22 = -NR
A33 = 'MXUD
B33 = -XU
KAI = -YDEL R
KBI = XDEL R
KCI = -XU

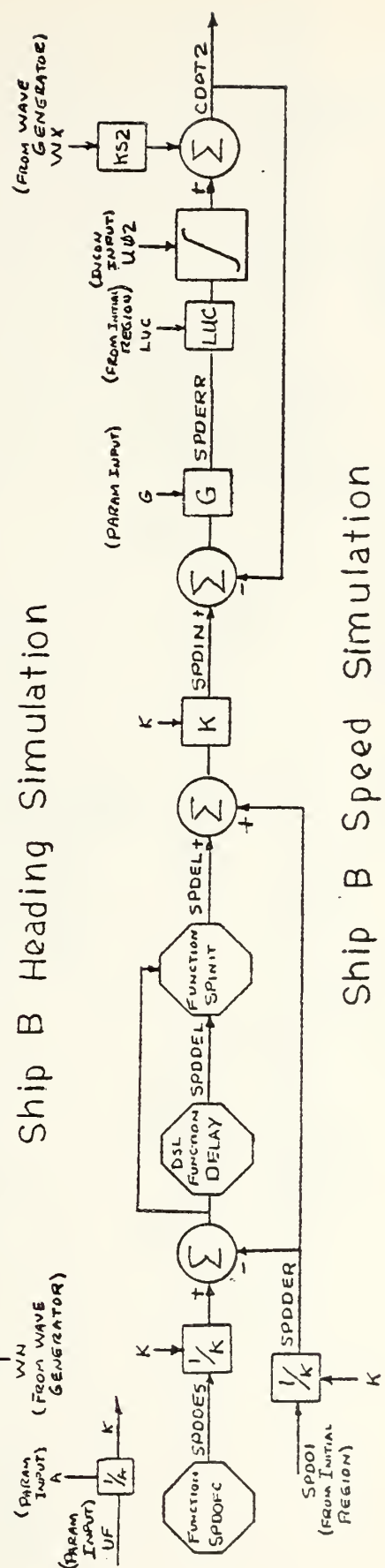


Ship A Speed Simulation

Ship A (Reference Ship) simulation

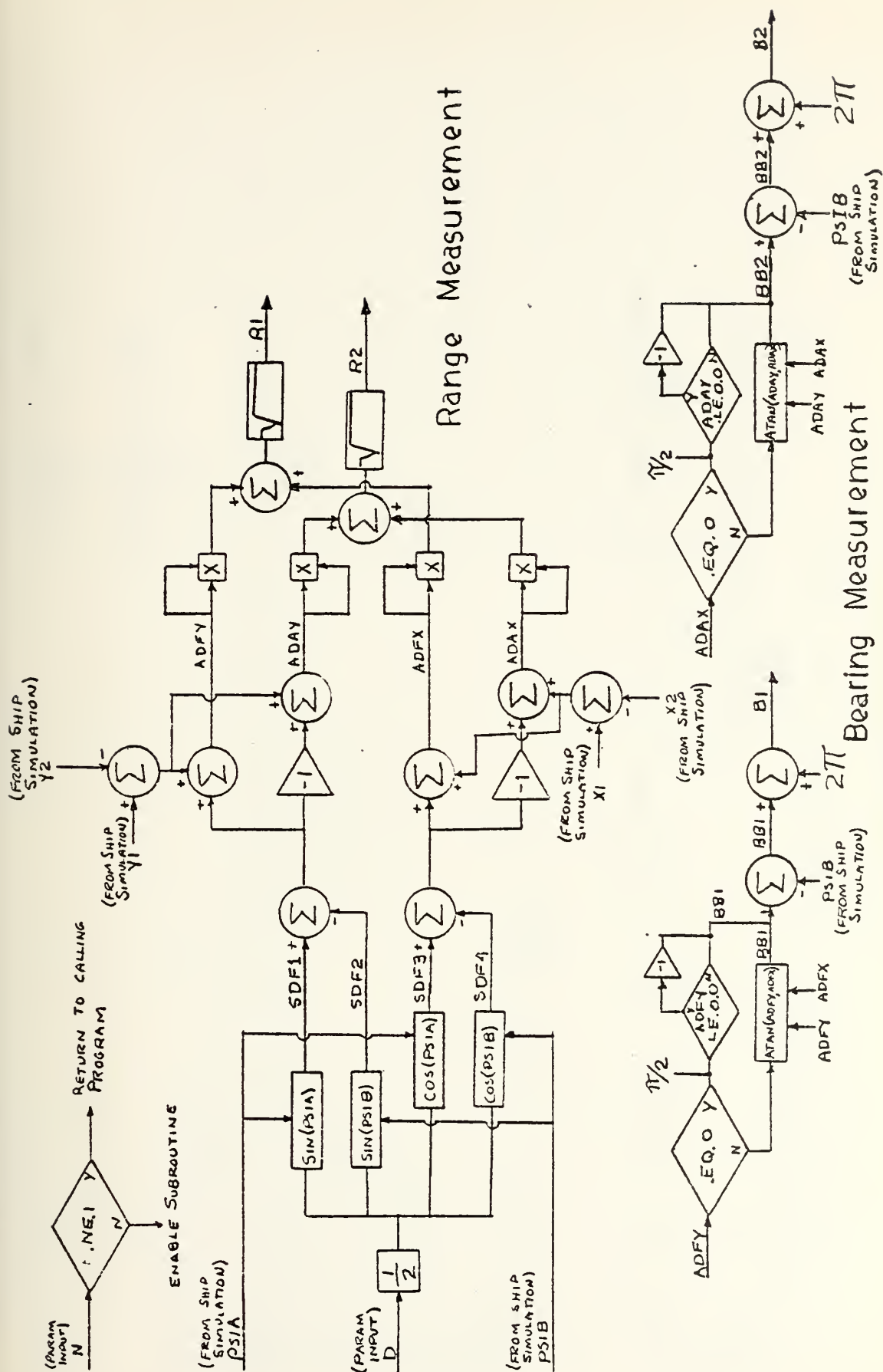


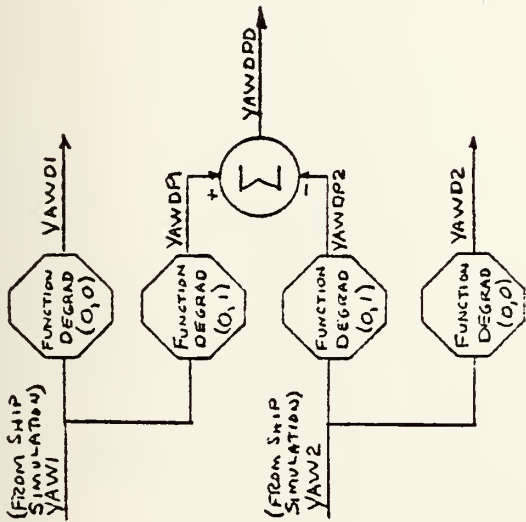
Ship B Heading Simulation



Ship B Speed Simulation

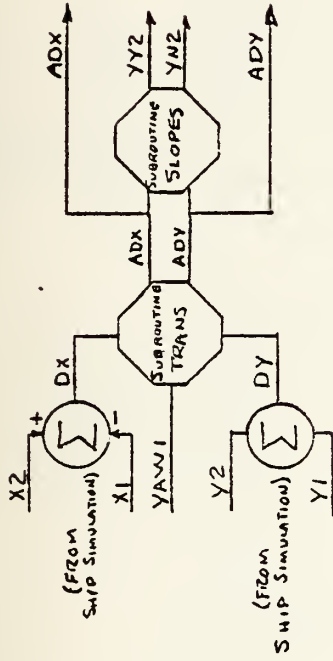
Ship B (Control Ship) Simulation



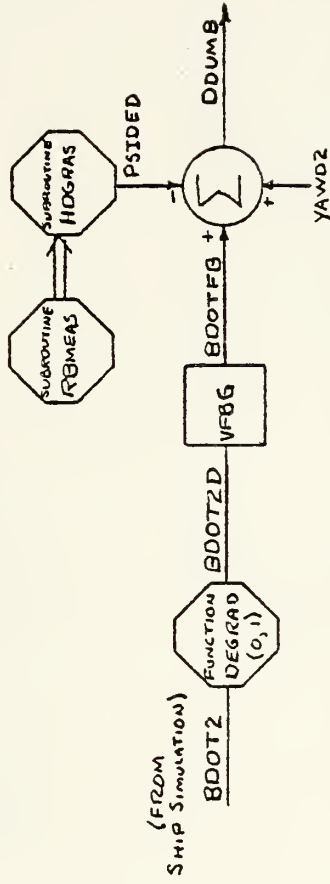


Yaw conversions

295

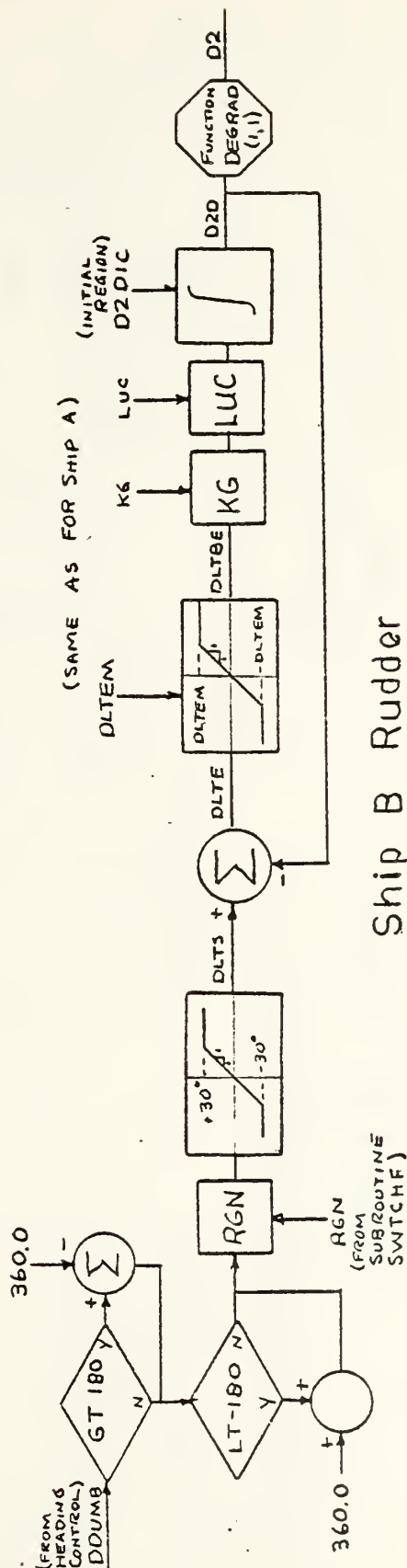
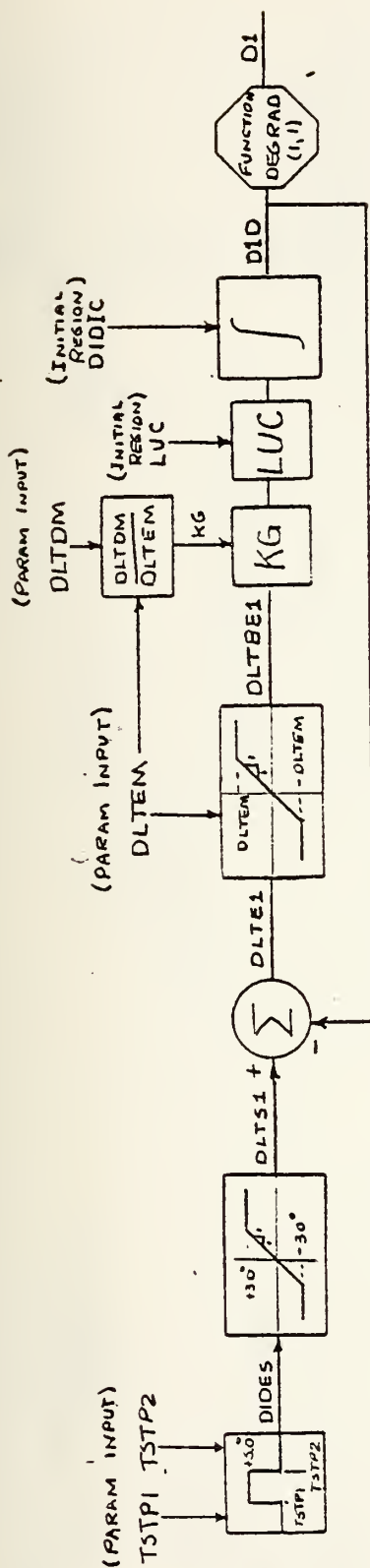


Coordinate Conversion

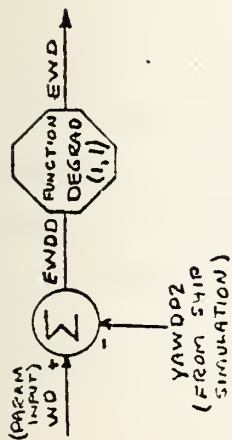


Feedback Loop

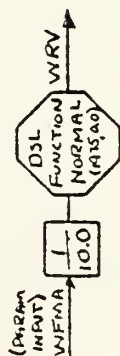
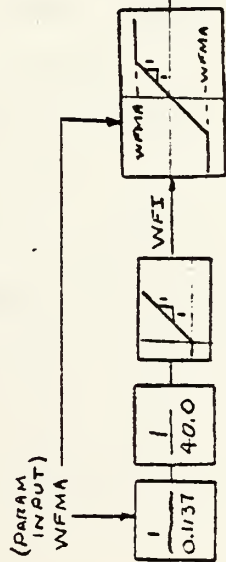
Auxiliary Functions



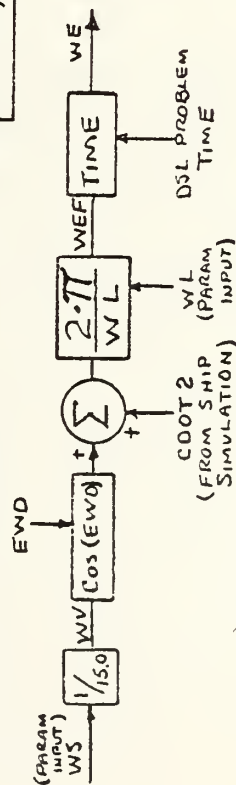
Rudder Modeling



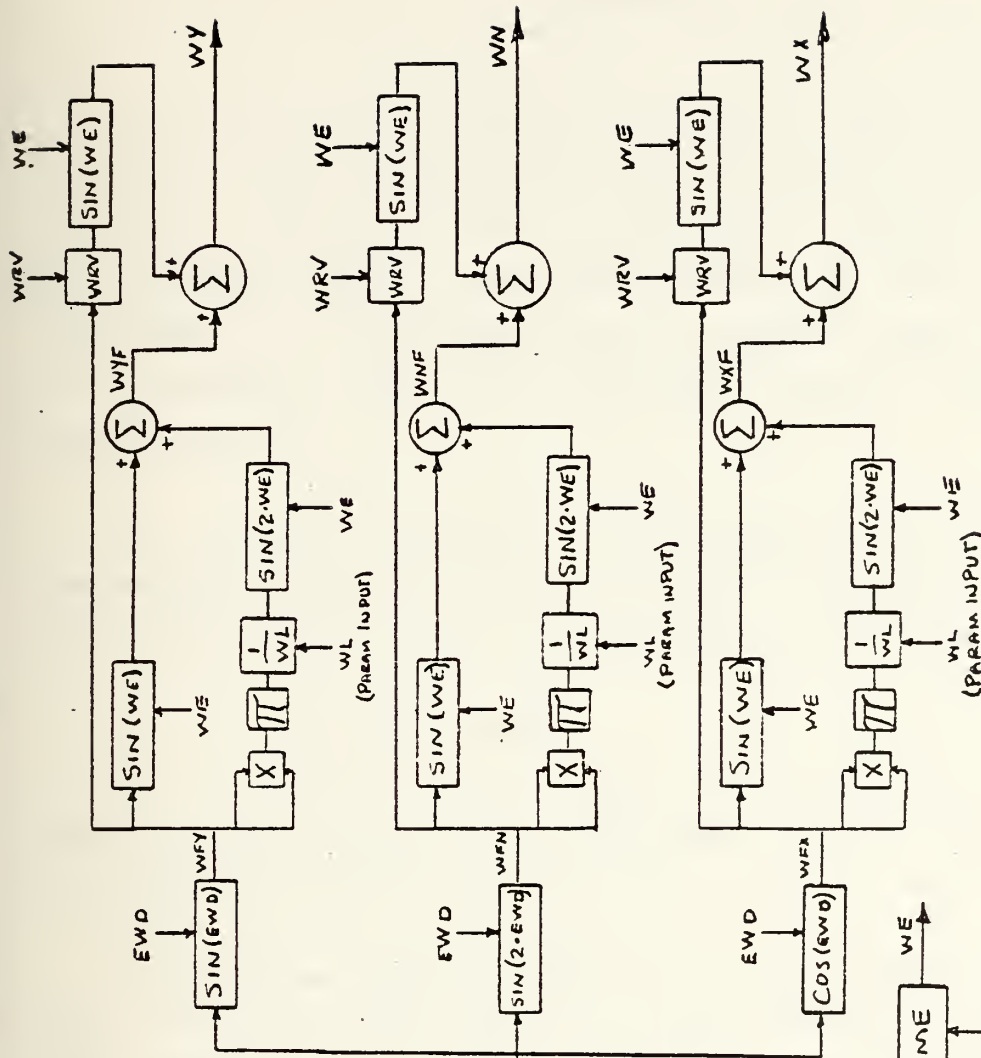
Wave Direction



Random Variable



Wave Encounter



Wave Components

Wave Generator

APPENDIX C

Throughout this thesis subroutine SLOPES has been used to output the interactive forces and moments between ships in the RAS situation. This subroutine, adapted from ref. 11, does not contain a complete picture of the circumstances envisioned. In particular, ship's speeds other than the 15 kt. operating point and different ship lengths are not accounted for.

As stated in chapter II, the speed modification factor can easily be applied for both ships at the same speed and other than 15 kts. with the following expression:

$$SPDP = CDOT^2$$

Ships replenishing with different lengths can also be incorporated as shown in ref. 1.

Subroutine FAMIC listed in this appendix incorporates these two ideas along with a better method of determining the interactive forces and moments. The curves of figures II-11 and II-12 were quantized every 50 feet of DX for all the DY curves shown. These points were then used in the NFGS XDS-930C digital computer and AGT-10 graphics terminal to obtain a family of best fit curves. The best fit criteria is based on the sum of the error squared at each quantized point (modified somewhat by this researcher's evaluation of best fit between points to eliminate spikes and other anomalies). The results of this curve fit process is summarized in tables C-1 and C-2, which includes tabulation of the best fit criteria. These polynomial coefficients are based on the DX distance and are coded in

Power	Y50 [YY(1)]	Y60 [YY(2)]	Y70 [YY(3)]	Y80 [YY(4)]	Y90 [YY(5)]	Y100 [YY(6)]
0	84.324	75.260	67.668	59.223	53.449	47.441
1	.364580E 2	.384289E 2	.305160E 2	.221400E 2	.174604E 2	.115702E 2
2	-.118950E 3	-.116865E 3	-.103683E 3	-.944770E 2	-.740983E 2	-.587039E 2
3	-.197311E 2	-.504085E 2	-.265717E 2	-.710687E 1	-.885819E 0	.165712E 2
4	.513437E 2	.684302E 2	.719316E 2	.103147E 3	.467865E 2	.242239E 2
5	.503561E 1	.466749E 2	.158656E 2	-.286063E 1	-.749374E 1	-.397052E 2
6	-.100337E 2	-.194633E 2	-.344841E 2	-.106332E 3	-.206166E 2	-.164427E 1
7	-.484175E 0	-.238722E 2	-.532674E 1	-.304098E 0	.488631E 1	.345891E 2
8	.750949E 0	-.398443E 0	.101864E 2	.688338E 2	.584863E 1	.677233E 0
9		.655075E 1	.877360E 0	.456701E 1	-.136413E 1	-.158999E 2
10		.181994E 1	-.158465E 1	-.221987E 2	-.941379E 0	-.255071E 1
11		-.916189E 0	-.556587E-1	-.347210E 1	.178935E 0	.405863E 1
12		-.428170E 0	.984583E-1	.160105E 1	.729225E-1	.131265E 1
13		.514317E-1		.114485E 1	-.901537E-2	-.543239E 0
14		.316613E-1		.965145E 0	-.176277E-2	-.256741E 0
15				-.179844E 0		.296983E-1
16				-.256001E 0		.177684E-1
17				.109745E-1		
18				.189745E-1		

Table C-1
Interactive Curve Fit Polynomial Coefficients

Power	Y110[YY(7)]	Y120[YY(8)]	Y130[YY(9)]	Y140[YY(10)]	Y150[YY(11)]
0	40.423	35.484	30.606	26.063	22.876
1	.142822E 2	.146142E 2	.138458E 2	.120761E 2	.108418E 2
2	-.533075E 2	-.458814E 2	-.425663E 2	-.341076E 2	-.196219E 2
3	-.103692E 2	-.158329E 2	-.139724E 2	-.150249E 2	-.127593E 2
4	.389319E 2	.398457E 2	.661640E 2	.574245E 2	.133189E 2
5	.875165E 1	.163639E 2	.863816E 1	.122510E 2	.974262E 1
6	-.200712E 2	-.242537E 2	-.800393E 2	-.702577E 2	-.695905E 1
7	-.502212E 1	-.953711E 1	-.274913E 1	-.333288E 1	-.335866E 1
8	.615393E 1	.820245E 1	.533407E 2	.488103E 2	.214837E 1
9	.154525E 1	.291771E 1	.211749E 1	-.946233E 0	.453764E 0
10	-.972521E 0	-.137319E 1	-.175113E 2	-.195409E 2	-.339142E 0
11	-.237633E 0	-.447954E 0	-.190223E 1	.766196E 0	.407256E-3
12	.607427E-1	.890809E-1	.163223E 1	.448663E 1	.211973E-1
13	.144010E-1	.272508E-1	.777504E 0	-.162184E 0	-.347399E-2
14			.560018E 0	-.548550E 0	
15			-.141969E 0	.116108E-1	
16			-.160683E 0	.276591E-1	
17			.960782E-2		
18			.120067E-1		

Table C-1
Interactive Curve Fit Polynomial Coefficients

Power	N50 [YN(1)]	N60 [YN(2)]	N70 [YN(3)]	N80 [YN(4)]	N90 [YN(5)]	N100 [YN(6)]
0	-37.329	-32.887	-29.637	-25.559	-23.064	-19.946
1	.396089E 2	.326330E 2	.242094E 2	.188936E 2	.173015E 2	.135031E 2
2	.402860E 2	.362146E 2	.371281E 2	.304472E 2	.328997E 2	.311296E 2
3	-.326885E 2	-.256484E 2	-.711639E 1	-.125164E 1	-.559773E 1	-.199635E 1
4	.397695E 1	.348686E 1	-.705366E 1	-.599694E 0	-.167065E 2	-.200747E 2
5	.847350E 1	.747495E 1	-.121721E 2	-.141799E 2	-.505714E 1	-.613206E 1
6	-.188304E 2	-.174840E 2	-.782624E 1	-.133875E 2	.626911E 1	.938014E 1
7	.334908E 0	-.236966E 1	.894936E 1	.915469E 1	.308497E 1	.324620E 1
8	.103131E 2	.102702E 2	.657382E 1	.943453E 1	-.179228E 1	-.279550E 1
9	-.136166E 1	.992335E 0	-.251129E 1	-.251460E 1	-.605389E 0	-.630851E 0
10	-.222623E 1	-.287062E 1	-.224627E 1	-.303990E 1	.301083E 0	.443308E 0
11	.647133E 0	-.217277E 0	.326965E 0	.332880E 0	.410884E-1	.438239E-1
12	.338223E-1	.397733E 0	.366197E 0	.478124E 0	-.206993E-1	-.281713E-1
13	-.132735E 0	.168090E-1	-.164232E-1	-.174528E-1		
14	.553291E-1	-.218380E-1	-.231377E-1	-.294675E-1		
15	.989063E-2					
16	-.574936E-2					

Table C-1
Interactive Curve Fit Polynomial Coefficients

Power	N110[YN(7)]	N120[YN(8)]	N130[YN(9)]	N140[YN(10)]	N150[YN(11)]
0	-16.682	-13.636	-10.637	-8.880	-6.955
1	.101849E 2	.103644E 2	.800782E 1	.603195E 1	.502563E 1
2	.223191E 2	.191665E 2	.104719E 2	.106262E 2	.784026E 1
3	-.108658E 0	-.867847E 1	-.587329E 1	-.222096E 1	-.261221E 1
4	-.296954E 1	-.102589E 2	.423064E 1	-.348743E 1	-.211257E 1
5	-.587218E 1	.793490E 1	.508267E 1	-.205426E 0	.551911E 0
6	-.100911E 2	-.357415E 1	-.106235E 2	.518022E 0	.292597E 0
7	.333381E 1	-.610692E 1	-.388765E 1	.102580E 0	-.213673E 0
8	.830859E 1	-.697142E 0	.702568E 1	-.198157E-1	-.779550E-1
9	-.891852E 0	.235015E 1	.146642E 1	-.592635E-2	.513869E-1
10	-.281000E 1	.575924E-1	-.225402E 1	-.269964E-2	.209270E-1
11	.124323E 0	-.420471E 0	-.258120E 0		-.396563E-2
12	.443430E 0	-.835171E-3	.354734E 0		-.211007E-2
13	-.718710E-2	.283053E-1	.172570E-1		
14	-.268301E-1		-.218733E-1		

Table C-1
Interactive Curve Fit Polynomial Coefficients

Curve Fit	Best Fit			Best Fit (Modified)		
	Order	$\sum e^2$	\bar{e}	Order	$\sum e^2$	\bar{e}
Y50	15	2.1664	0.307	8	70.345	1.749
Y60	15	2.1462	0.305	14	2.1703	0.307
Y70	15	0.41727	0.135	12	1.1915	0.228
Y80	15	0.71421	0.176	18	0.87588	0.195
Y90	15	1.2793	0.236	14	1.2981	0.238
Y100	16	1.1573	0.224	16	1.1573	0.224
Y110	15	1.2798	0.236	13	1.3400	0.241
Y120	15	0.39722	0.131	13	0.77148	0.183
Y130	15	0.54194	0.154	18	0.66737	0.170
Y140	15	0.77259	0.183	16	1.0620	0.215
Y150	15	0.26589	0.108	13	0.44726	0.139
N50	17	0.80547	0.187	16	0.80572	0.187
N60	18	0.63879	0.167	14	0.73449	0.179
N70	17	0.57433	0.158	14	0.59042	0.160
N80	14	0.43632	0.138	14	0.43632	0.138
N90	17	0.77685	0.184	12	0.83329	0.190
N100	15	0.48934	0.146	12	0.67948	0.172
N110	15	0.25701	0.106	14	0.59247	0.160
N120	15	0.29538	0.113	13	0.62067	0.164
N130	15	0.051807	0.047	14	0.064323	0.053
N140	15	0.13166	0.076	10	0.30835	0.116
N150	15	0.11837	0.072	12	0.16549	0.085
Avg.		0.71425	0.176		3.96169	0.252

Table C-2
Interactive Curve Fit Error Analysis

subroutine FAMIC as YY(1) thru YY(11) and YN(1) thru YN(11). An interpolation algorithm is used to determine the forces and moments at DY points between the curves of each family. Although all the computations are based on measurements from the control ship (ship #2), the interactive forces and moments are also computed for the reference ship (ship #1).

Figures C-1 and C-2 are the interactive forces and moments output to show comparison to figures II-11 and II-12. The speed of this run was the operating point of 15 kts. The ships are of equal length (527.8 feet).

Linear interpolation of the interactive curves for greater than 150 feet DY distance is accomplished from this 150 foot curve to a value of 0.0 at 200 feet. It therefore assumes no force and moment are present outside the 200 foot range. All forces and moments for DY distance of less than 50 feet are taken as that of the 50 foot curve. These two endpoints are by no means exact, but will suffice until more detailed data can be gathered. Another inexact endpoint is produced at the curve families limits of ± 550 feet. At these points, the forces and moments are forced to 0.0 since detailed data outside of these limits was not available. A side effect of this abrupt truncation will manifest itself in the instantaneous commencement of the forces and moments during the approach phase run. The endpoint variations in some of the curves of figures C-1 and C-2 are due in part to the curve fitting routine used, but mostly to the differences in computer precision. (curve fits were calculated on a 11 digit precision XDS 9300 while the curves were plotted on single precision 7 digit IBM 360/67)

As previously mentioned, the speed modification for other than the operating point of 15 kts. is only completely valid for the situation where both ships are at the same speeds. Since this thesis considered an approach phase

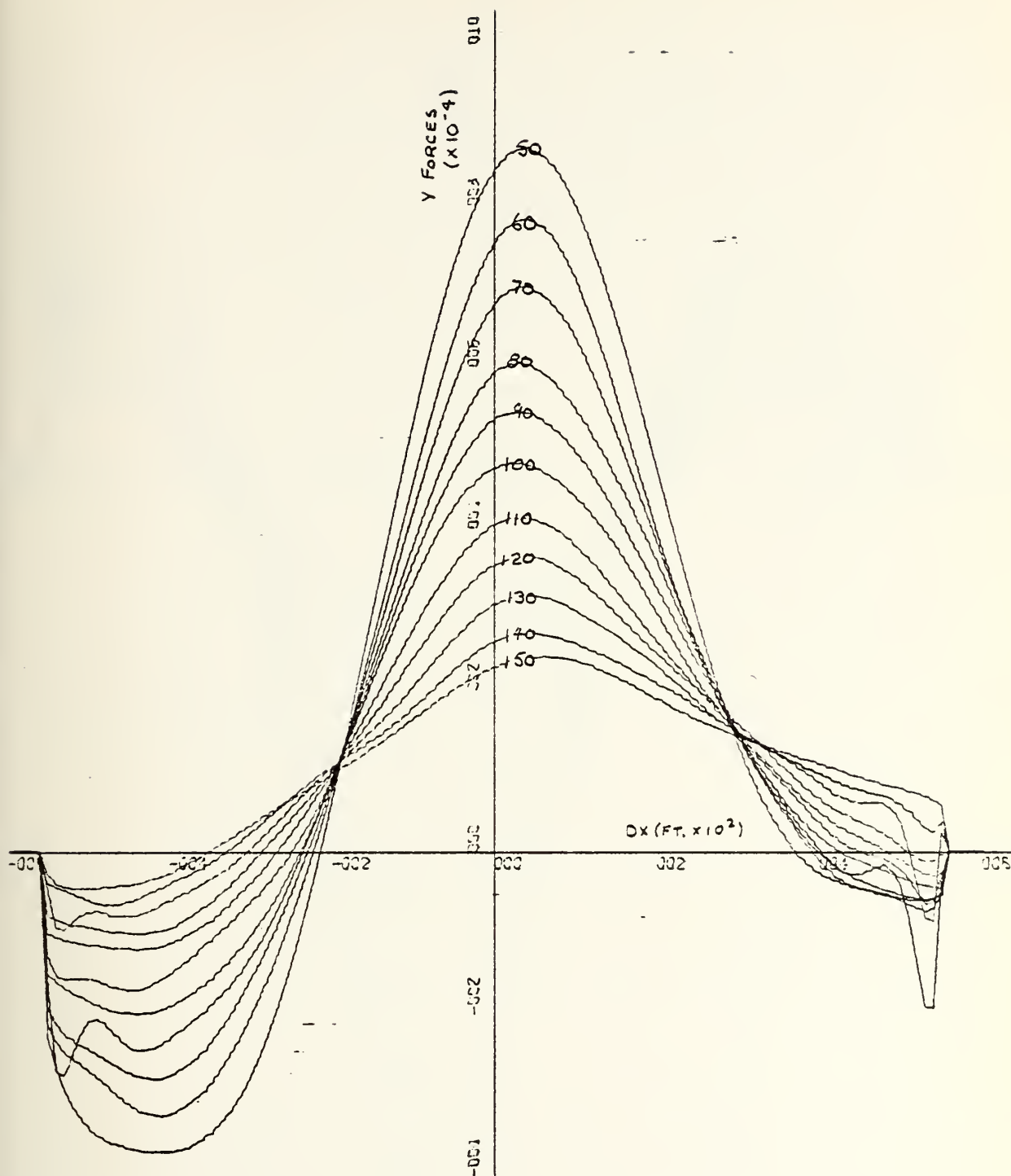


Figure C-1
Curve Fitted Interactive Y Forces

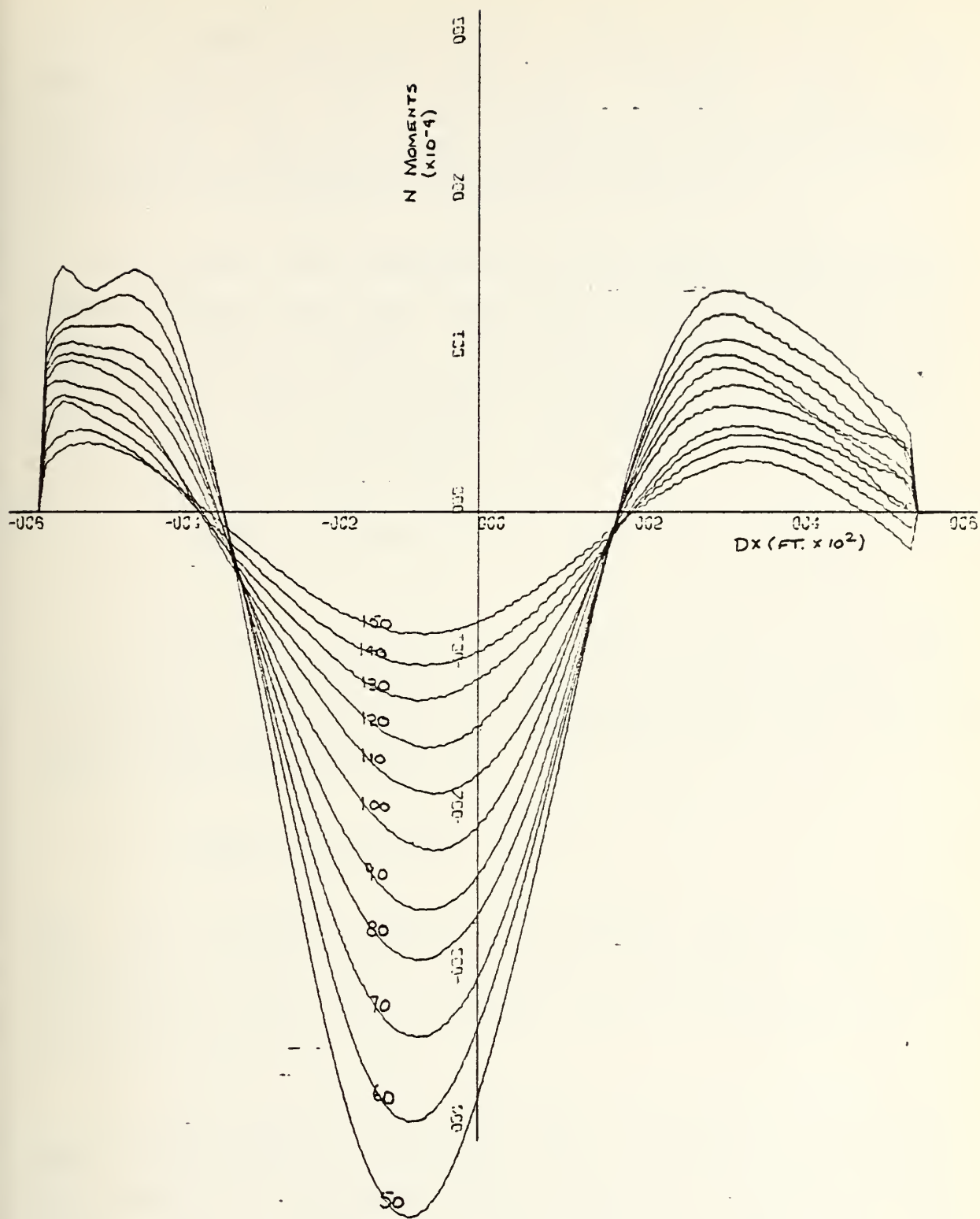


Figure C-2
Curve Fitted Interactive N Moments

where the control ship enters the interactive field at a speed quite different than the reference ship, some modification of the interactive effects should be considered. However, exact relationships are not available to compute the required modification factors.

To dispel any problems with the design of the heading control system, the worst case speed modification factor was chosen. This factor, in effect, considers that the interactive forces and moments are derived from the control ship. This is accomplished in subroutine FAMIC with the following fortran expression:

$$SFIF2 = CDOT2**2$$

As stated in chapter II, it is felt that it is more accurate to consider the interactive forces and moments to be modified by the speed of the reference ship, and can be coded in subroutine FAMIC as:

$$SFIF2 = CDOT1**2$$

With the scenario followed throughout this thesis, this expression would equate to unity throughout the RAS situation, since the reference ship is maintained at 1.0 normalized speed (15 kts.).

For the sake of error analysis, simulation of the worst case modification is performed. This gives rise to forces and moments 2.25 times what they were in the rest of this thesis during a portion of the approach phase when the normalized speed of the control ship is 1.5. Figures C-3 and C-4 show the interactive forces and moments for the approach phase of the simulation. The comparison plots which appear in chapter III as figures III-24 and III-25

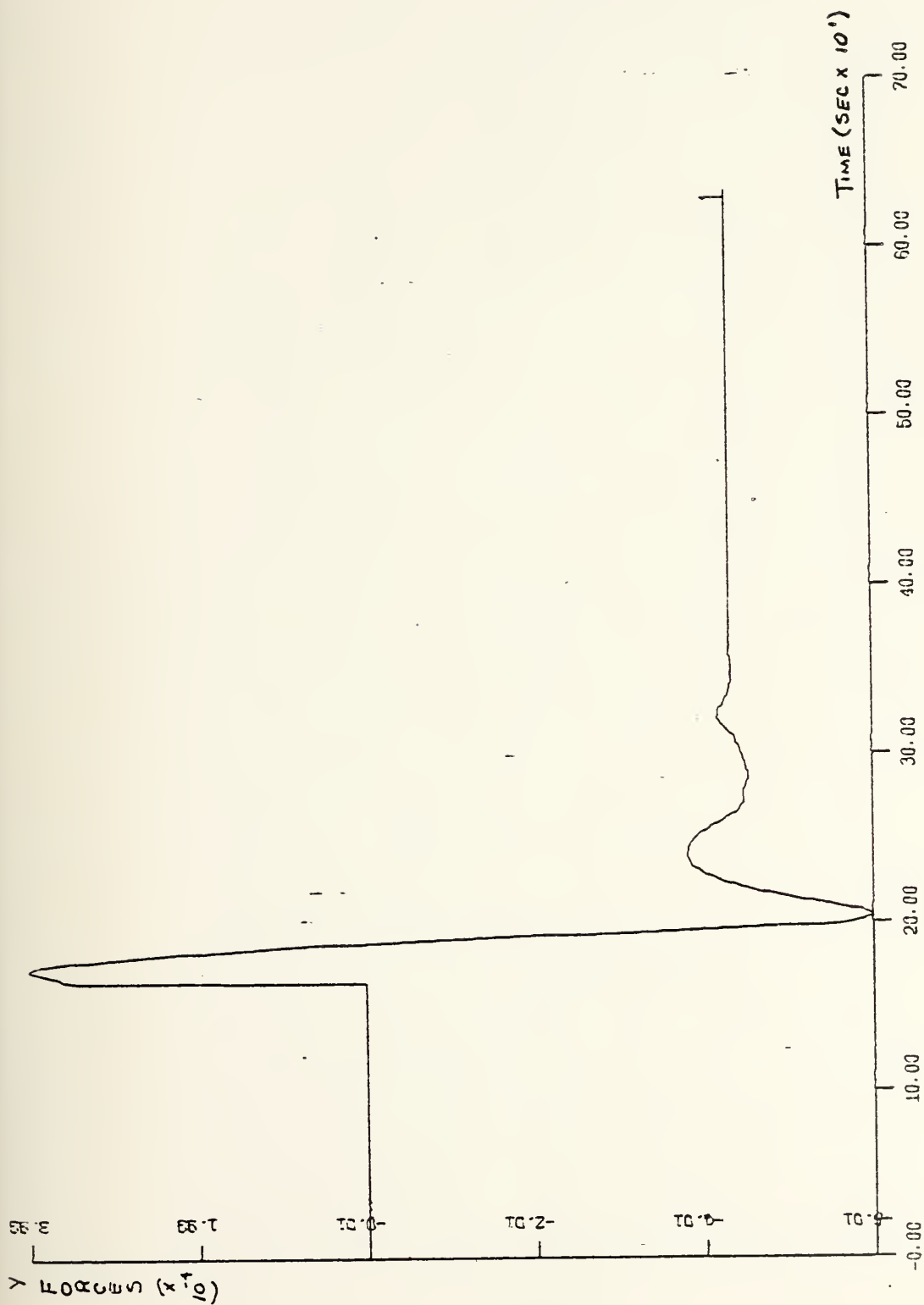


Figure C-3
Approach Phase Curve Fitted Y Forces

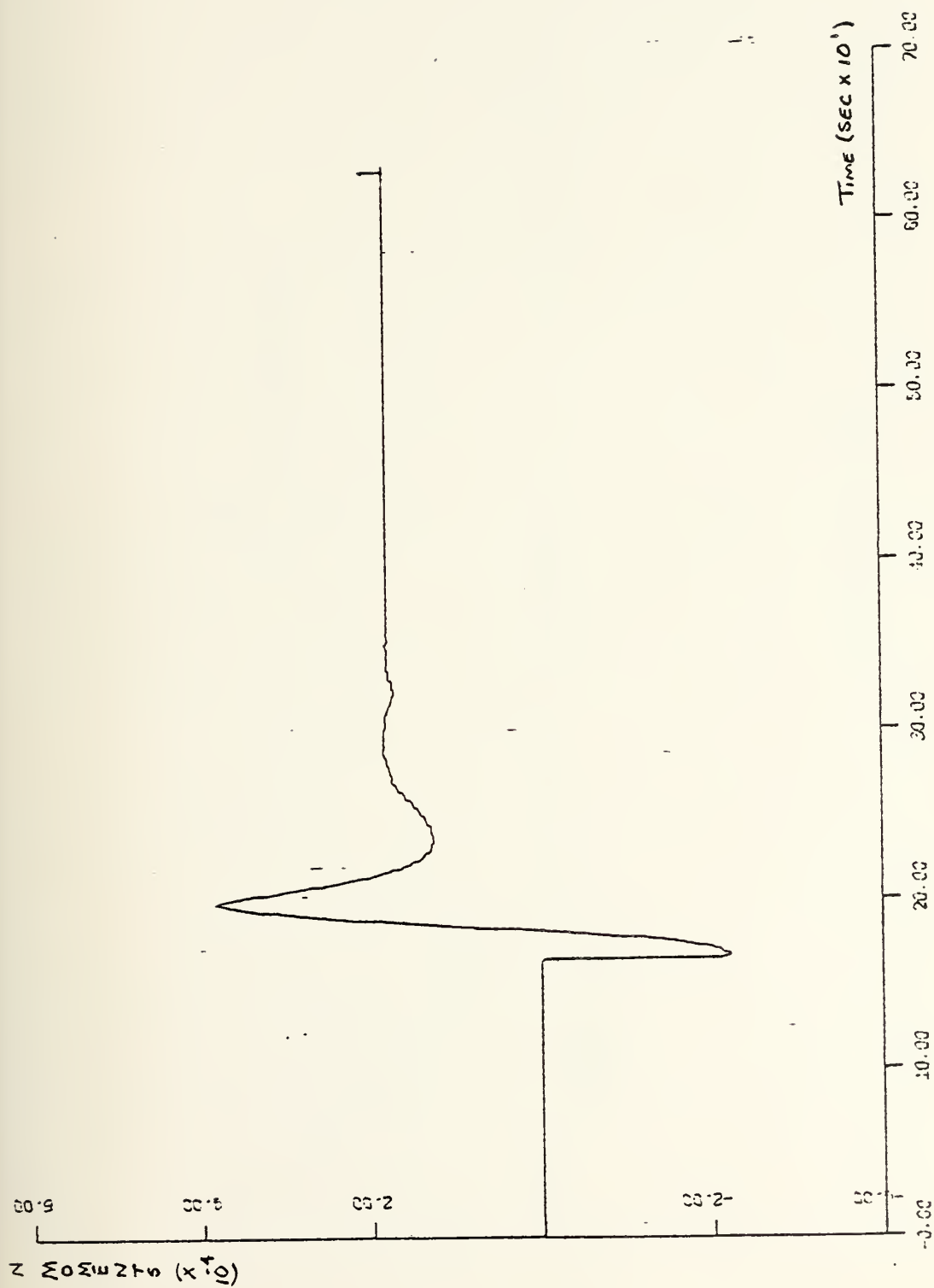


Figure C-4
Approach Phase Curve Fitted N Moments

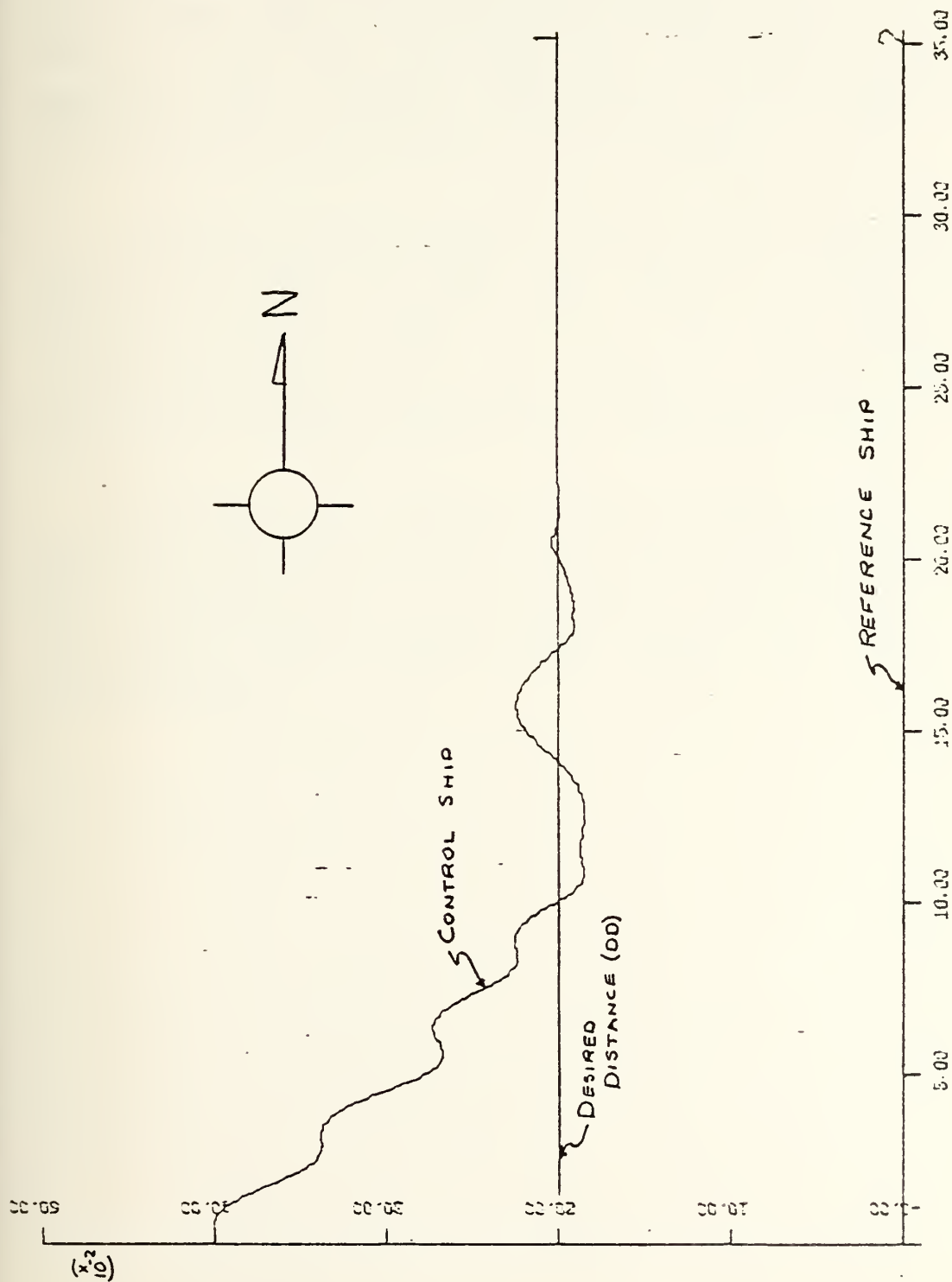


Figure C-5
Approach Phase Geographical Plot From Modified Interactive Effects

illustrate the extent of the changes. Most notable is the smoother output of subroutine FAMIC. This more realistically portrays the interactive effects in the RAS environment. Figure C-5 portrays the geographical plot which compares with figure III-26 without speed modification. Although differences exist, figure C-5 illustrates that the interactive effects speed modification factor for the worst case does not drastically alter the approach phase outcome. The heading control system design is still valid in the face of these changes.

For reference, figures C-6 and C-7 show the interactive forces and moments in the turn phase as calculated by subroutine FAMIC. Figure C-8 is the turn phase lateral distance plot produced. It can be seen from this illustration that the maximum excursion is 0.0056 normalized distance (2.96 feet), well within acceptable limits.

In summary, the designed control system will accommodate even the worst case modification of the interactive effects. This insensitivity to a large range of perturbations, makes this control system a more viable design for actual ship installation.

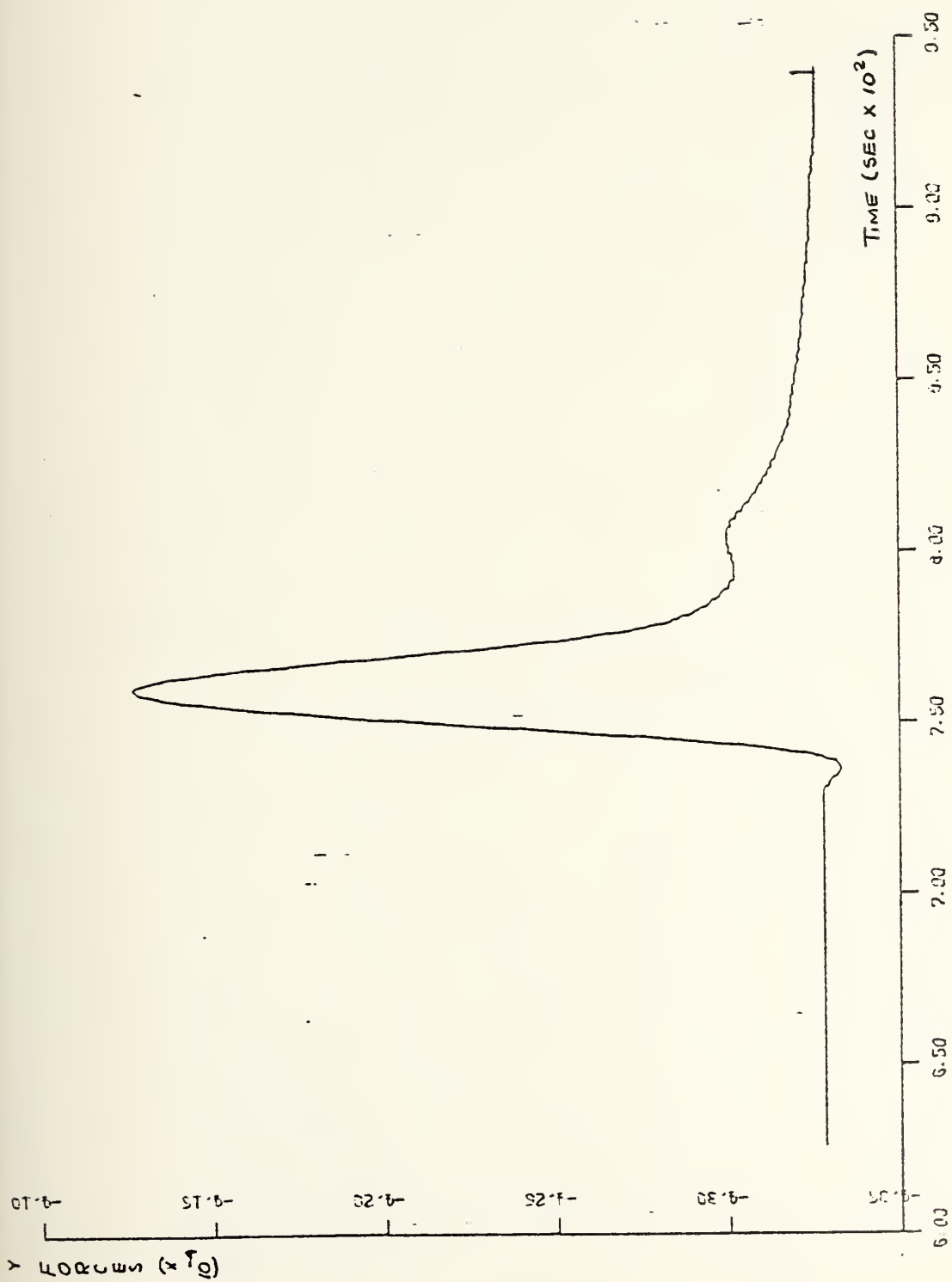
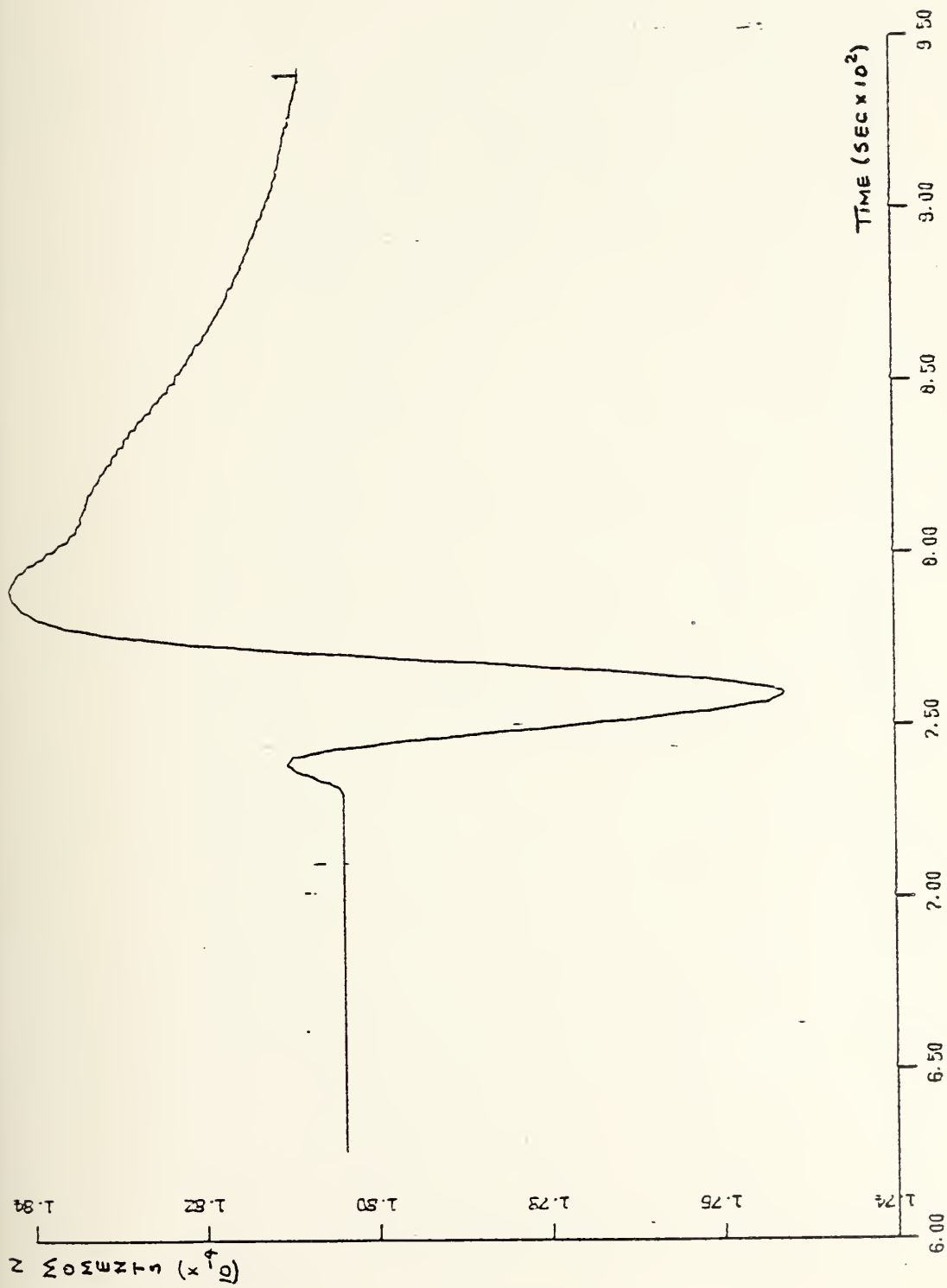


Figure C-6
Turn Phase Curve Fitted Y Forces



Figure, C-7
Turn Phase Curve Fitted N Moments

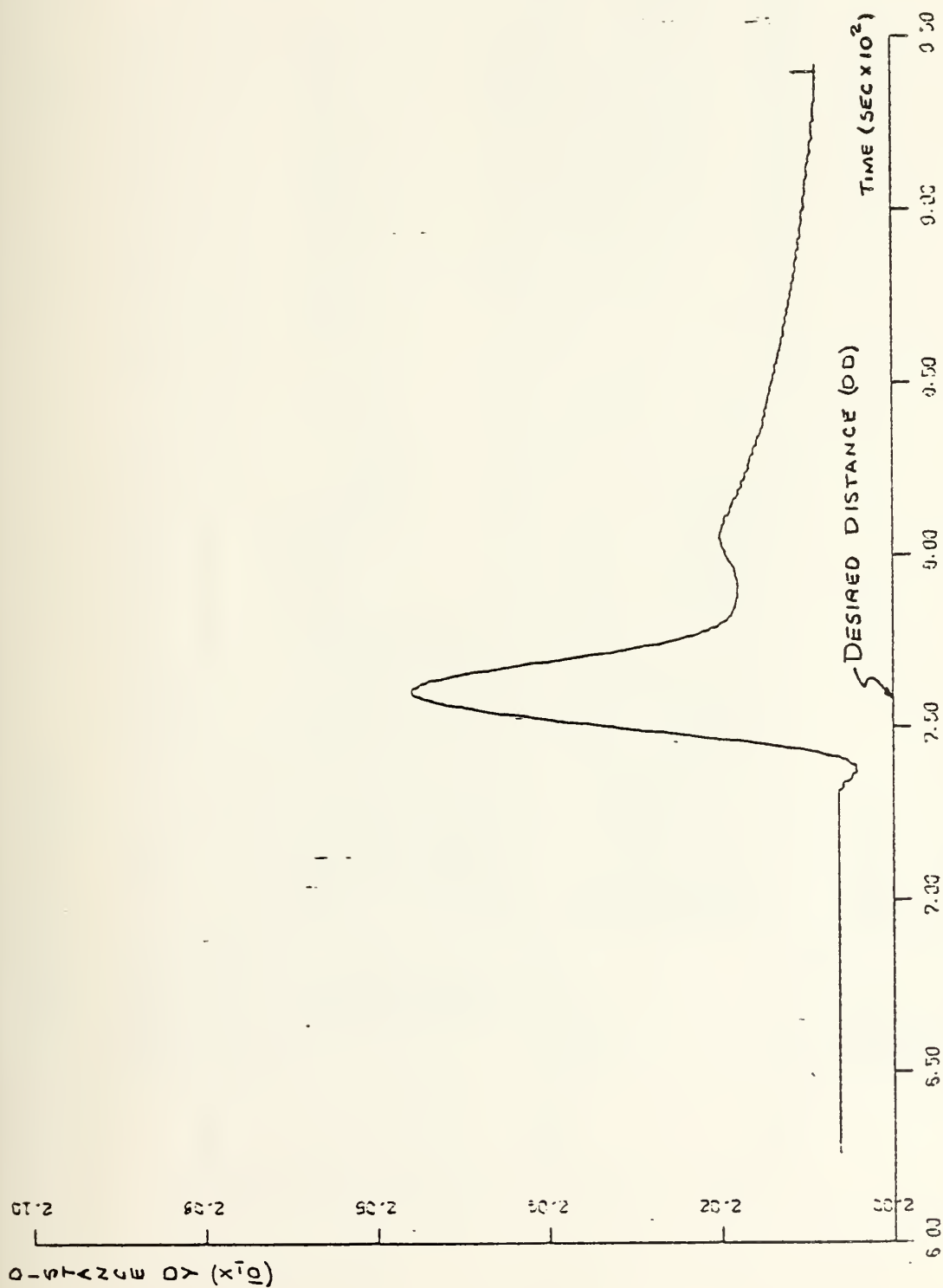


Figure C-8

Turn Phase Lateral Distance DY From Modified Interactive Effects

SUBROUTINE FAMIC

SUBROUTINE FAMIC (XL1,XL2,ADX,ADY,CDO11,CDO12,YY1,YY2,YN1,YN2)
 DIMENSION YY(12), YN(12)

DEFINITION OF TERMS:

XL1 = LENGTH OF SHIP #1 IN FEET
 XL2 = LENGTH OF SHIP #2 IN FEET
 ADX = LONGITUDINAL SEPARATION OF SHIPS (NONDIMENSIONALIZED)
 ADY = LATERAL SEPARATION OF SHIPS (NONDIMENSIONALIZED)
 CDO11 = NORMALIZED SPEED OF SHIP #1
 CDO12 = NORMALIZED SPEED OF SHIP #2
 YY1 = Y FORCE ON SHIP #1
 YY2 = Y FORCE ON SHIP #2
 YN1 = N MOMENT ON SHIP #1
 YN2 = N MOMENT ON SHIP #2
 X = LONGITUDINAL SEPARATION NORMALIZED TO CURVE FIT FACTOR

NOTE:

FOR STBD SIDE TO APPROACH OF SHIP #2 ON SHIP #1, ADY
 SHOULD BE NEGATIVE.
 FOR APPROACH WHEN SHIP #2 IS ASTERN OF SHIP #1 ADX SHOULD
 BE NEGATIVE.
 CONVERSES ARE ALSO TRUE - PORT SIDE TO = POSITIVE ADY AND
 FWD OF ALONGSIDE = POSITIVE.
 ADX AND ADY ARE REFERENCED TO SHIP #2.
 NORMALIZED SPEEDS BASED ON 15 KTS.

XL = XL2
 SPDP1 = CDO11**2
 SPDP2 = CDO12**2
 XL1 = XL1/XL2
 XL2 = 1.0
 N = 1

1 ACXL = XL*ADX
 X = XL*ADX/250.0
 IF (ABS(ADXL).GT.550.0) GO TO 6

PCLYNOMIAL POWER PRE-COMPUTATION


```

189319E+02*X4+0.875165E+01*X5-0.200712E+02*X6-0.502212E+01*X7+0.615FAMI 920
2353E+01*X8+0.154525E+01*X9-0.972521E+00*X10-0.237633E+00*X11+0.607FAMI 930
2427E-01*X12+0.144010E-01*X13 940
Y(8) = 35.484+0.146142E+02*X1-0.458814E+02*X2-0.158329E+02*X3+0.3FAMI 950
198457E+02*X4+0.163639E+02*X5-0.242537E+02*X6-0.953711E+01*X7+0.820FAMI 960
2245E+01*X8+0.291771E+01*X9-0.137319E+01*X10-0.447954E+00*X11+0.890FAMI 970
38C9E-01*X12+0.272508E-01*X13 980
Y(9) = 30.606+0.138458E+02*X1-0.425663E+02*X2-0.139724E+02*X3+0.6FAMI 990
161640E+02*X4+0.863816E+01*X5-0.800393E+02*X6-0.274913E+01*X7+0.533FAMI 1000
24C7E+02*X8+0.211749E+01*X9-0.175113E+02*X10-0.190223E+01*X11+0.163FAMI 1010
3223E+01*X12+0.777504E+00*X13+0.560018E+00*X14-0.141569E+00*X15-0.1FAMI 1020
460683E+00*X16+0.960782E-02*X17+0.120067E-01*X18 1030
Y(10) = 26.063+0.120761E+02*X1-0.341076E+02*X2-0.150249E+02*X3+0.4FAMI 1040
1574245E+02*X4+0.122510E+02*X5-0.702577E+02*X6-0.333288E+01*X7+0.48FAMI 1050
281C3E+02*X8-0.946233E+00*X9-0.195409E+02*X10+0.766156E+00*X11+0.44FAMI 1060
38663E+01*X12-0.162184E+00*X13-0.548550E+00*X14+0.116108E-01*X15+0.4FAMI 1070
427659E-01*X16 1080
Y(11) = 22.876+0.108418E+02*X1-0.196219E+02*X2-0.127593E+02*X3+0.21FAMI 1090
1133189E+02*X4+0.974262E+01*X5-0.695905E+01*X6-0.335866E+01*X7+0.21FAMI 1100
24837E+01*X8+0.453764E+00*X9-0.339142E+00*X10+0.407256E-03*X11+0.21FAMI 1110
31973E-01*X12-0.347359E-02*X13 1120

```

EQUATIONS OF N MOMENTS

```

YN(1) = -37.329+0.356089E+02*X1+0.402860E+02*X2-0.326885E+02*X3+0.10FAMI 1160
1357655E+01*X4+0.847350E+01*X5-0.188304E+02*X6+0.3347508E+00*X7+0.10FAMI 1170
238231E+02*X8-0.136166E+01*X9-0.222623E+01*X10+0.647133E+00*X11+0.33FAMI 1180
4574936E-02*X16 1190
YN(2) = -32.887+0.326330E+02*X1+0.362146E+02*X2-0.256484E+02*X3+0.10FAMI 1200
1348686E+01*X4+0.747495E+01*X5-0.174840E+02*X6-0.236966E+01*X7+0.10FAMI 1210
22702E+02*X8+0.992335E+00*X9-0.287062E+01*X10-0.217277E+00*X11+0.39FAMI 1220
37733E+00*X12+0.168090E-01*X13-0.218380E-01*X14 1230
YN(3) = -29.637+0.242094E+02*X1+0.371281E+02*X2-0.711639E+01*X3-0.65FAMI 1240
1705366E+01*X4-0.121721E+02*X5-0.782624E+01*X6+0.894936E+01*X7+0.36FAMI 1250
27382E+01*X8-0.251129E+01*X9-0.224627E+01*X10+0.326965E+00*X11+0.36FAMI 1260
36197E+00*X12-0.164232E-01*X13-0.231377E-01*X14 1270
YN(4) = -25.559+0.188936E+02*X1+0.304472E+02*X2-0.125164E+01*X3-0.94FAMI 1280
1599694E+00*X4-0.141799E+02*X5-0.133875E+02*X6+0.915469E+01*X7+0.94FAMI 1290
233453E+01*X8-0.251460E+01*X9-0.303990E+01*X10+0.332880E+00*X11+0.47FAMI 1300
38124E+00*X12-0.174528E-01*X13-0.294675E-01*X14 1310
YN(5) = -23.064+0.173015E+02*X1+0.328997E+02*X2-0.559773E+01*X3-0.17FAMI 1320
1167065E+02*X4-0.505714E+01*X5+0.626911E+01*X6+0.308497E+01*X7-0.17FAMI 1330
29228E+01*X8-0.605389E+00*X9+0.301083E+00*X10+0.410884E-01*X11-0.20FAMI 1340
36593E-01*X12 1350
YN(6) = -19.946+0.135031E+02*X1+0.311296E+02*X2-0.195635E+01*X3-0.27FAMI 1360
1200747E+02*X4-0.613206E+01*X5+0.938014E+01*X6+0.324620E+01*X7-0.27FAMI 1370
29550E+01*X8-0.630851E+00*X9+0.443308E+00*X10+0.438239E-01*X11-0.28FAMI 1380

```



```

3 1713E-01*X12      101849E+02*X1+0.223191E+02*X2-0.108658E+00*X3-0.      FAMI11400
   YN(7) = -16.682+0.587218E+00*X5-0.100911E+02*X6+0.333381E+01*X7+0.83      FAMI11410
12 56954E+01*X8-0.891852E+00*X9-0.281000E+01*X10+0.124323E+00*X11+0.44      FAMI11420
20 659E+01*X12-0.718710E-02*X13-0.268301E-01*X14      FAMI11430
33 4330E+00*X12-0.636+0.103644E+02*X1+0.191665E+02*X2-0.867847E+01*X3-0.      FAMI11440
   YN(8) = -13.636+0.103644E+02*X1+0.191665E+02*X2-0.867847E+01*X3-0.      FAMI11450
17 142589E+02*X4+0.793490E+01*X5+0.357415E+01*X6-0.610652E+01*X7-0.69      FAMI11460
27 142E+00*X8+0.235015E+01*X9+0.575924E-01*X10-0.420471E+00*X11-0.83      FAMI11470
35 171E-03*X12+0.283053E-01*X13      FAMI11480
   YN(9) = -10.637+0.800782E+01*X1+0.104719E+02*X2-0.587329E+01*X3+0.      FAMI11490
14 23064E+01*X4+0.508267E+01*X5-0.106235E+02*X6-0.388765E+01*X7+0.70      FAMI11500
22 568E+01*X8+0.146642E+01*X9-0.225402E+01*X10-0.258120E+00*X11+0.35      FAMI11510
34 734E+00*X12+0.172570E-01*X13-0.218733E-01*X14      FAMI11520
   YN(10) = -8.880+0.603195E+01*X1+0.106226E+02*X2-0.222096E+01*X3-0.      FAMI11530
13 48743E+01*X4-0.205426E+00*X5+0.518022E+00*X6+0.102580E+00*X7-0.19      FAMI11540
28 117E-01*X8-0.955+0.502563E-02*X9-0.269964E-02*X10      FAMI11550
28 11257E+01*X4+0.551911E+00*X5+0.784026E+01*X2-0.261222E+01*X3-0.      FAMI11560
29 550E-01*X8+0.513869E-01*X9+0.209270E-01*X10-0.396563E-02*X11-0.21      FAMI11570
31 007E-02*X12      FAMI11580
   ACYA = ABS(ADY)      FAMI11590
   ADYL = XL2*ADYA      FAMI11600
   XII = (ADYL-40.0)/10.0      FAMI11610
   I = XII      FAMI11620
   IF (XII.LT.1.0.OR.I.LT.1) GO TO 4      FAMI11630
   IF (XII.GT.11.0.OR.I.GT.11) GO TO 5      FAMI11640
   IF (XII.LT.XII) GO TO 2      FAMI11650
   IF (XII.GT.XII) GO TO 3      FAMI11660
   YVI = YN(I)      FAMI11670
   YNI = YN(I)      FAMI11680
   GC TO 7      FAMI11690
   YVI = YN(I)+(YV(I+1)-YV(I))*((XII-XI))      FAMI11700
   YNI = YN(I)+(YN(I+1)-YN(I))*((XII-XI))      FAMI11710
   GC TO 7      FAMI11720
   YVI = YN(I)-(YV(I)-YV(I-1))*((XII-XI))      FAMI11730
   YNI = YN(I)-(YN(I)-YN(I-1))*((XII-XI))      FAMI11740
   GC TO 7      FAMI11750
   YVI = YN(I)      FAMI11760
   YNI = YN(I)      FAMI11770
   GC TO 7      FAMI11780
   YVI = YN(I)      FAMI11790
   YNI = YN(I)      FAMI11800
   GO TO 7      FAMI11810
   IF (ADYL.GT.200.0) GO TO 6      FAMI11820
   YVI = YN(11)*(1.0+(150.0-ADYL)/50.0)      FAMI11830
   YNI = YN(11)*(1.0+(150.0-ADYL)/50.0)      FAMI11840
   GC TO 7      FAMI11850
   YVI = 0.0      FAMI11860
   YNI = 0.0      FAMI11870

```

C


```

YN1 = 0.0
YN2 = 0.0
RETURN
7 IF (N*EC.2) GO TO 8
  YY1*SPDP2*XL P2*1.0E-05
  YN2 = -YN1*SPDP2*XL P2*1.0E-05
  N = N+1
  XL = -XL2
  GO TO 1
8 YY1 = YY1*SPDP1*XL P1*1.0E-05
  YN1 = YN1*SPDP1*XL P1*1.0E-05
  IF (ADY.LT.0.0) GO TO 9
  RETURN
9 YY1 = -YY1
  YY2 = -YY2
  YN1 = -YN1
  YN2 = -YN2
  RETURN
END

```

```

FAMI11880
FAMI11890
FAMI11900
FAMI11910
FAMI11920
FAMI11930
FAMI11940
FAMI11950
FAMI11960
FAMI11970
FAMI11980
FAMI11990
FAMI12000
FAMI12010
FAMI12020
FAMI12030
FAMI12040
FAMI12050
FAMI12060

```


COMPUTER PROGRAM #1

This program incorporates the ship dynamics of two identical Mariner hulls. These hulls are superimposed in space to allow for comparison of the effects contributed to rudder modeling differences. In this particular run a step and ramp rudder were compared in chapter II.

Another benefit of this program is to set up the two identical ships required for the RAS simulations in chapter III. Basically, verification of the models in three degrees of freedom is accomplished for the Mariner hull chosen.

The plots produced in this run are shown in figures II-2 and II-3.

COMPUTER PROGRAM #1

```

//UHRINTF1 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL INPUT DD *
* LINEAR RESPONSE OF THE MARINER - RAMP VS STEP RUDDER COMPARISON
* LINEAR RESPONSE OF THE MARINER - RAMP VS STEP RUDDER COMPARISON
TITLE TRAPZ
INTEGER NPLOT=1
CONST NPLOT=1
* HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,I2NRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XJ=-0.0012,YRD=-0.00027
CONST YDEL=-0.0027,NDEL=-0.00126,XDEL=-0.0
* INITIAL CONDITIONS
INCON X01=0.0,Y01=0.0,X02=0.0,Y02=0.0
INITIAL
* CALCULATION OF THE COEFFICIENTS
D1=0.0
D2=0.0
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=MYR
A12=-NV
B12=-NV
A22=I2NRD
B22=-NR
A33=MXUD
B33=-XU
KAI=-YDEL
KB1=NDEL
KC1=XDEL
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
DRC=3.1415926/180.
LUC=20.84765
DERIVATIVE
* SIMULATION SHIP A
IF1=KAI*D1
IF21=KB1*D1

```



```

IF31=KC1*DI+NC1
I11=-B11*ADOT1-B21*BDO11+IF11
I21=-B12*ADOT1-B22*BDO11+IF21
I31=-B33*CDOT1+IF31
ADDOT1=(I11*A22-I21*A21)/D
BDDOT1=(I21*A11-I11*A12)/D
CDDOT1=I31/A33
ADOT1=INTGRL(0.,ADDOT1)
BDO11=INTGRL(0.,BDDOT1)
CDO11=INTGRL(1.0,CDDOT1)
B1=INTGRL(0.,BDO11)
XDOT1=CDO11*COS(B1)-ADOT1*SIN(B1)
YDOT1=CDO11*SIN(B1)+ADOT1*COS(B1)
X1=INTGRL(X01,XDOT1)
Y1=INTGRL(Y01,YDOT1)
YAW1=B1
SWAY1=Y1
SURGE1=X1
SIMULATION SHIP B
IF12=KB1*D2
IF22=KB1*D2
IF32=KC1*D2+NC2
I12=-B11*ADOT2-B21*BDO12+IF12
I22=-B12*ADOT2-B22*BDO12+IF22
I32=-B33*CDOT2+IF32
ADDOT2=(I12*A22-I22*A21)/D
BDDOT2=(I22*A11-I12*A12)/D
CDDOT2=I32/A33
ADOT2=INTGRL(0.,ADDOT2)
BDO12=INTGRL(0.,BDDOT2)
CDO12=INTGRL(1.0,CDDOT2)
B2=INTGRL(0.,BDO12)
XDOT2=CDO12*COS(B2)-ADOT2*SIN(B2)
YDOT2=CDO12*SIN(B2)+ADOT2*COS(B2)
X2=INTGRL(X02,XDOT2)
Y2=INTGRL(Y02,YDOT2)
YAW2=B2
SWAY2=Y2
SURGE2=X2
DYNAMIC REGION
RUDDER RESPONSE INPUT
D1=0.2618*STEP(2.0)
D2=RAMP(2.0)*DELRM*DRC
IF(D2*GE.0.2618) D2=0.2618
SWAYD=SWAY2-SWAY1
SURGED=SURGE2-SURGE1
YAWD1=YAW1*GRDC
YAWD2=YAW2*RDC

```

*

*


```

YAWDD=YAWD1-YAWD2
D1D=D1*RDC
D2D=D2*RDC
DDD=D1D-D2D
ATIME=LUC*TIME

SAMPLE 0.04, ATIME, YAWD1, D1D, YAWD2, D2D, YAWDD, DDD
PRINT
CONTRL FINIM=30., DELT=0.04, DELS=0.04
PRPLOT ONLY
CALL DRWG(1,1, SURGE1, SWAY1)
CALL DRWG(1,2, SURGE2, SWAY2)
CALL DRWG(2,1, ATIME, YAWDD)
CALL DRWG(3,1, ATIME, DDD)

TERMINAL
CALL ENDRW(NPLOT)

END
STOP
//PLOT.SYSIN DD *

0.0      1.0      -5.0      1.0      8.0      6.0
0.0      100.0    -6.0      1.0      8.0      6.0
INSERT TWO /* CARDS HERE
7.0      5.0

```

5 5 4

COMPUTER PROGRAM #2

This program models a practical rudder response for a mariner ship type. The rudder limits (stops) are set at ± 30 degrees and the rate of response is limited to ± 2 degrees/sec. A scale factor (LUC) is introduced to modify the response to match real time of the mariner hull chosen.

Twelve passes thru the program are accomplished to conform to different sets of initial conditions and final desired rudder conditions. The plots produced in this run are shown in figures II-5 and II-6.

COMPUTER PROGRAM #2

```
//UHRINTF2 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=1
//EXEC DSL
//DSL INPUT DD *
* RAS RUDDER CONTROL RUN TF2 - PRACTICAL RUDDER RESPONSE
* RAS RUDDER CONTROL RUN TF2 - PRACTICAL RUDDER RESPONSE
TITLE TRAPZ
INTEGER NPLOT,CURVE
CONST NPLOT=2
CONST DLTDM=2.0,DLTEM=7.0
PARAM CURVE=1
PARAM D2DDES=30.0
PARAM D2DIC=-30.0
INITIAL
  KG=DLTDM/DLTEM
  LUC=20.84765
DERIVATIVE
  DLTS=LIMIT(-30.0,30.0,D2DDES)
  DLTE=DLTS-D2D
  DLTE=LIMIT(-DLTEM,DLTEM,DLTE)
  D2D=INTGRL(D2DIC,KG*DLTE*LUC)
  ACTUAL TIME CONVERSION
  ATIME=TIME*LUC
*
SAMPLE FINTIM=1.7,DELT=0.04,DELS=0.04
CONTRL PRINT 0.04,ATIME,DLTS,DLTE,DLTE,D2D,D2DDES,D2DIC
PRPLOT ONLY
  CALL DRWG(1,CURVE,ATIME,D2D)
TERMINAL
  WRITE(6,100)D2DIC,D2DDES
  100 FORMAT(/,' LAST RUN IS FOR INITIAL RUDDER=',F10.5,' DESIRED RUDDER=
  1 R=',F10.5)
  D2DDES=D2DDES-5.0
  D2DIC=D2DIC+5.0
  CURVE=CURVE+1
  IF(NPLOT.EQ.1) D2DIC=0.0
  IF(CURVE.EQ.7) GO TO 1
  GO TO 2
  1 CURVE=1
    D2DIC=0.0
    D2DDES=30.0
    CALL ENDRW(NPLOT)
  2 CALL RERUN
END
```


END
STOP
//PLOT.SYSIN DD *

0.0 5.0 -30.0 10.0 7.0 6.0

5

0.0
INSERT TWO /* CARDS HERE

5.0 0.0 5.0 7.0 6.0

5

COMPUTER PROGRAM #3

This program models a reduced order (first order) gas turbine propulsion plant for an input-output relationship. The program does not scale the plant to the mariner hull used. This was done when introduced into the main simulation program first listed as computer program #8.

The time delay (P) is assisted in initialization by a dual feed into the system; one thru the delay itself and one directly into SPDIN. The program can be modified to compare a family of curves by introducing the following sequence into the TERMINAL region:

```
INTEGER NUMB
```

```
IF (NCUR.EQ.NUMB) CALL ENDEW(NPLOT)
```

```
IF (NCUR.NE.NUMB) CALL RERUN
```

```
NCUR = NCUR + 1
```

where NUMB is the number of curves desired (less than or equal to 10) which is set with a PARAM statement. The comparison is done on the conditions set in the terminal region [i.e. decrement or increment the system gains (eg. $G = G + 0.02$)].

The plot produced by this run is shown as part of figure II-10.

COMPUTER PROGRAM #3

```

//UHRINTF3 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL.INPUT DD *
TITLE SPEED CONTROL - FIRST ORDER FIT
INTEG RKSTFX
INCON UIC=20.0
INTEGER NPLOT,NCUR
PARAM NPLOT=1
PARAM UF=21.73
PARAM A=22.0
PARAM P=0.0
PARAM NCUR=1
PARAM G=0.092
INITIAL
K=UF/A
WRITE(6,100) G,NCUR,K
100 FORMAT(//,1 THE FOLLOWING RUN FOR POLE=--,F10.5/,23X,'NCUR=',
113,/,23X,'K=',F10.5,/)
DERIVATIVE
SPDDER=20.25
SPDDES=1.75*STEP(10.0)
SPDDEL=DELAY(7,P,SPDDES)
SPDIN=K*SPDDEL+K*SPDDER
SPDEKR=(SPDIN-SPDOUT)*G
SPDOUT=INTGRL(UIC,SPDERR)
DYNAMIC
IF (TIME.GT.9.0) P=4.88
SAMPLE
CONTROL FINTIM=320.0,DELT=0.8,DELS=0.8
PRINT 1.6,SPDDES,SPDDEL,SPDIN,SPDERR,SPDOUT,P
TERMINAL
CALL ENDRW(NPLOT)
END
STOP
//PLCT.SYSIN DD *
0.0 40.0 20.0 0.4 8.0 5.0
INSERT TWO /* CARDS HERE

```


COMPUTER PROGRAM #4

This program models a simplified wave simulation composed of two superimposed sinusoids (fundamental and second harmonic) and a small random wave. The model is inherently scaled to the mariner nondimensional characteristics. Introduction of these waves is accomplished in computer program #7. Subroutine DEGRAD is shown in appendix A.

The sea state force plots in the dimensions of the three degrees of freedom produced by this run is shown in figures II-17 thru II-22.

COMPUTER PROGRAM #4

```

//UHRINTF4 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=2
// EXEC DSL
//DSL. INPUT DD *
* TITLE WAVE PERTURBATION SIMULATION
  INTGER NPLOT
  PARAM NPLOT=6
  PARAM YAWDP2=0.0
  PARAM CDOT2=1.5
  PARAM LUC=20.84765
  PARAM WS=5.0
  PARAM WD=015.0
  PARAM WL=0.5
  PARAM WFMA=0.1137
  ATIME=LUC*TIME
  WFI=(WFMA/(0.1137*40.0))*RAMP(0.0)
  WF=LIMIT(-WFMA,WFMA,WFI)
  WRV=NORMAL(1975,0.0,WFMA/10.0)
  WV=WS/15.0
  EWDD=WD-YAWDP2
  EWD=DEGRAD(1,1,EWDD)
  WEF=2.0*3.1415926*(CDOT2+WV*COS(EWD))/WL
  WE=WEF*TIME
  WFY=WF*SIN(EWD)
  WFN=WF*SIN(2.0*EWD)
  WFX=WF*WV*COS(EWD)
  WYF=WF*WV*SIN(EWD)+(3.1415926*WV**2/WL)*SIN(2.0*WE)
  WNF=WF*WV*SIN(WE)+(3.1415926*WV**2/WL)*SIN(2.0*WE)
  WXF=WF*WV*SIN(WE)+(3.1415926*WV**2/WL)*SIN(2.0*WE)
  WY=WF*WV*WV*WV*SIN(WE)
  WN=WNF+WRV*WV*WV*SIN(WE)
  WX=WXF+WRV*WV*WV*SIN(WE)
  WED=DEGRAD(0,1,WE)

SAMPLE 0.04,ATIME,WY,WYF,WN,WNF,WX,WXF,WED,WRV
PRINT ONLY
PRPLOT FINT IM=5.0,DELT=0.04,DELS=0.04
CONTROL CALL DRWG(1,1,ATIME,WX)
        CALL DRWG(1,2,ATIME,WY)
        CALL DRWG(1,3,ATIME,WN)
TERMINAL
WRITE(6,100) WS,WD,WL,WFMA,CDOT2
100 FORMAT(//,' LAST RUN FOR WS=',F10.5,/,14X,'WD=',F10.5,/,14X,'WL=',

```



```

1 F10.5/,14X,'WFMA=',F10.5/,14X,'CDOT2=',F10.5,////)
  CALL ENDRW(NPLOT)
END
PARAM WL=1.0
END
PARAM WL=1.5
END
PARAM WD=030.0
PARAM WL=0.5
END
PARAM WL=1.0
END
PARAM WL=1.5
END
STOP
FORTRAN
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
//PLOT.SYSIN DD *

```

7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4

INSERT TWO /* CARDS HERE

COMPUTER PROGRAM #5

This program uses the mariner hull model first introduced in computer program #1 and the control system designed in chapter III to simulate the approach phase of RAS. The subroutines and functions that are to be inserted from appendix A can also be done in object code by changing the word FORTRAN to OBJECT and placing pre-compiled decks in the same locations. In fact, due to the long length of subroutine SLOPES, this must be done to be able to run the simulation with the DSL default job control language (JCL) presently installed at the Naval Postgraduate School IBM 360/67.

The plots produced by this run are shown in figures III-7 thru III-13. By changing the gains and introducing the following code, the plots of figures III-14 thru III-19 are produced:

$$DIDES = 5.0*STEP(8.0) - 5.0*STEP(9.0)$$

COMPUTER PROGRAM #5

```

//UHRINTF5 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=4
// EXEC DSL
//DSL.INPUT DD *
* TITLE RAS RUDDER CONTROL - APPROACH PHASE
* INTEG RKSFY
* INTEG NFLOT
* CONST NPLT=1
* CONST N,IS
* CONST N=1,RD=1.0
* CONST IS=1,DD=0.2
* PARAM RSENS=1.86642,WTSENS=2.3869,RGN=23.4185,VFBG=4.35162
* CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
* CONST MYVD=0.015,MYR=0.0051,I2NRD=0.00068,MXUD=0.0085
* CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
* CONST YDEL=-0.0027,NDEL=-0.00126,XDEL=-0.0
* CONST DLDM=2.0,DLTEM=7.0
* INITIAL SEPERATION
* INCON X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
* INITIAL CONDITIONS
* INCON YAW01=0.0
* INCON Y01=0.,YY2=0.,YN1=0.,YN2=0.
* INCON U01=1.0,U02=1.5
INITIAL
DY0=Y02-Y01
DX0=X02-X01
CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
CALL SLOPES(ADX,ADY,Y01,YY2,YN1,YN2)
CALCULATION OF THE COEFFICIENTS
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=MYR
A12=-NVD
B12=-NV
A22=IZNFD
B22=-NR
A33=MXUD
B33=-XU

```



```

KAI=-YDELR
KBI=NDELR
KCI=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
CRC=3.1415926/180.
LUC=20.84765
KG=DLTDM/DLTEM
D2D=0.0
DERIVATIVE
*
REFERENCE SHIP RUDDER CONTROL
DIDES=0.0
DLTSI=LIMIT(-30.0,30.0,DIDES)
DLTEI=DLTSI-DID
DLTBEI=LIMIT(-DLTEM,DLTEM,DLTEI)
DID=INTGRL(DIDIC,KG*DLTBEI*LUC)
DI=DEGRAD(1,1,DID)
DX=X2-X1
DY=Y2-Y1
SIMULATION SHIP A
*
IF11=KAI*D1
IF21=KBI*D1
IF31=KCI*D1+NC1
I11=-B11*ADOT1-B21*BDOOT1+IF11
I21=-B12*ADOT1-B22*BDOOT1+IF21
I31=-B33*CDOT1+IF31
ADDOT1=(I11*A22-I21*A21)/D
BCDOT1=(I21*A11-I11*A12)/D
CDDOT1=I31/A33
ADOT1=INTGRL(0.,ADDOT1)
BCOT1=INTGRL(0.,BCDOT1)
CDOCT1=INTGRL(U01,CDDOT1)
BY1=INTGRL(0.,BDOOT1)
XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(X01,XDOT1)
Y1=INTGRL(Y01,YDOT1)
YAW1=BY1
SWAY1=Y1
SURGE1=X1
SIMULATION SHIP B
*
IF12=KAI*D2+YY2
IF22=KBI*D2+YN2
IF32=KCI*D2+YC2
I12=-B11*ADOT2-B21*BDOOT2+IF12
I22=-B12*ADOT2-B22*BDOOT2+IF22
I32=-B33*CDOT2+IF32

```



```

ADDOT2=(I12*A22-I22*A21)/D
BDDOT2=(I22*A11-I12*A12)/D
CDDOT2=I32/A33
ADOT2=INTGRL(0.,ADDOT2)
BCOT2=INTGRL(0.,BDDOT2)
COT2=SPDCTR(ADX,U01,U02)
BY2=INTGRL(BY02,BDOT2)
XDOT2=CDDOT2*COS(BY2)-ADDOT2*SIN(BY2)
YDOT2=CDDOT2*SIN(BY2)+ADDOT2*COS(BY2)
X2=INTGRL(X02,XDOT2)
Y2=INTGRL(Y02,YDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2

NOSORT
YAWD1=DEGRAD(0,0,YAW1)
YAWDP1=DEGRAD(0,1,YAW1)
YAWD2=DEGRAD(0,0,YAW2)
YAWDP2=DEGRAD(0,1,YAW2)
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...)
PSIDED,WT,DA,AID,BID,B2D,WTSENS,DD,RD)
BCOT2D=DEGRAD(0,1,BDOT2)
BDOTFB=VFBG*BDOT2D
DDUMB=YAWD2-PSIDED+BDOTFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS=LIMIT(-30.0,30.0,DDUMB*RGD)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)

SORT
RUDDER PART OF OBJECT FUNCTION
DTRAN=TIME*ABS(D2)
ROBJ=INTGRL(0.0,DTRAN)
DISTANCE PART OF OBJECT FUNCTION
DISTE=TIME*10.0*ABS(DD-ADY)
DCBJ=INTGRL(0.0,DISTE)
OBJECT FUNCTION
OBJ=ROBJ+DOBJ
ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)
EXTERNAL FORCES ACTING BETWEEN SHIPS

```



```

CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
FORMAT(1, '*****SEPARATION LESS THAN 25 FEET - COLLISION*****')
100 ACTUAL TIME CONVERSION (SEC)
* ATIME=LUC*TIME

SAMPLE FINT IM=20., DELT=0.04, DELS=0.04
CONTRL 0.04,X1,X2,DA,AID,PSIDFD,Y1,Y2,R1,BID,PSIADD,YAWD1,YAWD2,R2,...
PRINT B2D,PSIDED,ATIME,D2D,ADX,DID,YY2,BDOTFB,DLTS,ADY,DLTS1,...

PRPLOT YN2,CDO1,CDOT2,OBJ
CNLY
CALL DRWG(1,1,ATIME,YAWDP2)
CALL DRWG(1,2,ATIME,YAWDP1)
CALL DRWG(2,1,ATIME,YY2)
CALL DRWG(3,1,ATIME,YN2)
CALL DRWG(4,1,SURGE2,SWAY2)
CALL DRWG(4,2,SURGE1,SWAY1)
CALL DRWG(5,1,ATIME,ADY)
CALL DRWG(6,1,ATIME,DLTS)
CALL DRWG(6,2,ATIME,D2D)

TERMINAL
IF(IS.EQ.1) WRITE(6,101)
101 FORMAT(1, 'THIS RUN IS FOR A PORT SIDE TO APPROACH')
IF(IS.EQ.0) WRITE(6,102)
102 FORMAT(1, 'THIS RUN IS FOR A STBD SIDE TO APPROACH')
CALL ENDRW(NPLOT)

END
STOP
FORTRAN
INSERT FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//G.FT06F001 DD SYSOUT=0,SPACE=(4,1)
//PLCT.SYSIN DD *

```

4

7.0 5.0

4

7.0 5.0

4

7.0 5.0

7

0.0 0.1

7.0 5.0

4

7.0 5.0

4

7.0 5.0

INSERT TWO /* CARDS HERE

COMPUTER PROGRAM #6

This program combines the approach and turn phases of computer program #5. The added subroutine is a result of simulation requirements to switch between adaptive gains.

This run produced the plots of figures III-22 thru III-34. By substituting the initial conditions of table III-3, this program produced the plots of figures III-35 thru III-64.

COMPUTER PROGRAM #6

```

//UHRINTF6 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL INPUT DD *
* RAS RUDDER CONTROL - APPROACH PHASE FOLLOWED BY TURN PHASE
TITLE RAS RUDDER CONTROL - APPROACH PHASE FOLLOWED BY TURN PHASE
INTEG RKSF
INTEG NFLOT
CONST NPLOT=2
INTEG N,IS
CONST N=1,RD=1.0
* SET IS FOR SIDE OF APPROACH IS=1 PORT, IS=0 STBD
* SET DD FOR DESIRED FINAL LATERAL SEPARATION DESIRED
IS=1,DD=0.2
CONST RSENS=1.86642
PARAM WTSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0,TSTP2=36.0
* HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227,NV=-0.00351,NVD=-0.000197
CONST MYVD=0.015,MYR=0.0051,I2NRD=0.00068,MXUD=0.0085
CONST YV=-0.01243,XU=-0.0012,YRD=-0.00027
CONST YDELR=-0.0027,NDELRL=-0.00126,XDELRL=0.0
CONST DLDM=2.0,DLTEM=7.0
* INITIAL SEPERATION
* SET IC FOR APPROACH TESTING
X01=5.0,Y01=0.0,X02=0.0,Y02=0.4
* INITIAL CONDITIONS
INCCN YAW01=0.0
INCON Y01=0.0,Y02=0.0,YN1=0.0,YN2=0.0
INCON U01=1.0,U02=1.5
INITIAL
DY0=Y02-Y01
DX0=X02-X01
CALL TRANS(YAW01,DX0,DY0,ADX,ADY)
CALL SLOPES(ADX,ADY,Y01,Y02,YN1,YN2)
CALCULATION OF THE COEFFICIENTS
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=MYR

```



```

A12=-NVD
B12=-NV
A22=IZNRD
B22=-NR
A33=MXUD
B33=-XU
KAI=-YDEL R
KBI=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
LUC=20.84765
KG=DLTDM/DLTEM
DIDIC=0.0
D2=DEGRAD(1,1,D2D)
BYG2=0.0
DERIVATIVE
DIDES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
DLTSI=LIMIT(-30.0,30.0,DIDES)
DLTEI=DLTSI-DID
DLTBEI=LIMIT(-DLTEM,DLTEM,DLTEI)
DID=INTGRL(DIDIC,KG*DLTBEI*LUC)
DI=DEGRAD(1,1,DID)
DX=X2-X1
DY=Y2-Y1
SIMULATION SHIP A
IF11=KAI*DI
IF21=KBI*DI+NC1
I11=-B11*ADOT1-B21*BDOOT1+IF11
I21=-B12*ADOT1-B22*BDOOT1+IF21
I31=-B33*CDOOT1+IF31
ADDOOT1=(I11*A22-I21*A21)/D
BDCOOT1=(I21*A11-I11*A12)/D
CDDOOT1=I31/A33
ACOT1=INTGRL(0.,ADDOOT1)
BCOT1=INTGRL(0.,BDDOOT1)
CCOT1=INTGRL(UO1,CDDOOT1)
BY1=INTGRL(0.,BDOOT1)
XCOT1=CDOOT1*COS(BY1)-ADOT1*SIN(BY1)
YDCT1=CDOOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(XO1,XDOOT1)
Y1=INTGRL(YO1,YDOOT1)
YAW1=BY1
SWAY1=Y1
SURGE1=X1

```

*


```

*
SIMULATION SHIP B
IF12=KA1*D2+YY2
IF22=KB1*D2+YN2
IF32=KC1*D2+NC2
I12=-B11*ADOT2-B21*BDOIT2+IF12
I22=-B12*ADOT2+IF32
I32=-B33*CDOT2+IF32
ACDOT2=(I12*A22-I22*A21)/D
BDDOT2=(I22*A11-I12*A12)/D
CDDOT2=I32/A33
ADOT2=INTGRL(0.,ADDOT2)
BCOT2=INTGRL(0.,BDDOT2)
CDOT2=SPDCTR(ADX,U01,U02)
BY2=INTGRL(BY02,BDOT2)
XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
X2=INTGRL(X02,XDOT2)
Y2=INTGRL(Y02,YDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2

NOSORT
YAWD1=DEGRAD(0,0,YAW1)
YAWDPI=DEGRAD(0,1,YAW1)
YAWD2=DEGRAD(0,0,YAW2)
YAWDP2=DEGRAD(0,1,YAW2)
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
CALL HDGRAS(N,IS,RI,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...,
PSIDED,WT,DA,AID,B1D,B2D,WTSENS,DD,RD)
BCOT2D=DEGRAD(0,1,BDOT2)
BCOTFB=VFBG*BDOIT2D
DDUMB=YAWD2-PSIDED+BDOITFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS=LIMIT(-30.0,30.0,DDUMB*RGD)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLTEM,DLTE)
D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)

SORT
DSTE=ABS(DD-ADY)
OBJ=INTGRL(0.0,DSTE)
DYNAMIC REGION
*
ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)

```



```

*      EXTERNAL FORCES ACTING BETWEEN SHIPS
      CALL SLOPES(ADX,ADY,Y1,Y2,YN1,YN2)
      IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
100    FFORMAT(, ****SEPARATION LESS THAN 25 FEET - COLLISION****)
*      ACTUAL TIME CONVERSION (SEC)
      ATIME=LUC*TIME
      AAI=(BB1+BB2)/2.0
      CALL SWITCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)

SAMPLE  FINTIM=30.,DELT=0.04,DELS=0.04
CONTRL  0.20,X1,X2,DA,AID,PSIDFD,Y1,Y2,R1,BID,PSIADD,YAWD1,YAWD2,R2,...
PRINT   B2D,PSIDED,ATIME,D2D,ADX,DID,YY2,BDOTFB,DLTS,ADY,DLTS1,...

PRPLCT  YN2,CDOOT1,CDOOT2,OBJ,RSENS,VFBG
        CNLY
        CALL DRWG(1,1,ATIME,YAWDP2)
        CALL DRWG(1,2,ATIME,YAWDP1)
        CALL DRWG(2,1,ATIME,YY2)
        CALL DRWG(3,1,ATIME,YN2)
        CALL DRWG(4,1,SURGE2,SWAY2)
        CALL DRWG(4,2,SURGE1,SWAY1)
        CALL DRWG(5,1,ATIME,ADY)
        CALL DRWG(6,1,ATIME,DLTS)
        CALL DRWG(6,2,ATIME,D2D)

TERMINAL
101    IF(IS.EQ.1) WRITE(6,101)
      IF(, THIS RUN IS FOR A PORT SIDE TO APPROACH' )
102    IF(IS.EQ.0) WRITE(6,102)
      IF(, THIS RUN IS FOR A STBD SIDE TO APPROACH' )
      CALL ENDRW(NPLOT)
      CALL CONTIN
      FINTIM=45.0

END
STOP
FORTRAN
INSERT  SUBROUTINE SWITCH FROM APPENDIX A HERE
INSERT  FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT  FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT  SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT  SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT  SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT  SUBROUTINE SLOPES FROM APPENDIX A HERE
//FLCT.SYSIN DD *

```

4

4

7.0 5.0

7.0 5.0

4 7 4 4 4 4 4 7 4 4

7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0

0.0 0.1

-2.0 0.5

INSERT TWO /* CARDS HERE

COMPUTER PROGRAM #7

This program combines the calm sea simulation of computer program #6 with the wave simulation of computer program #4 to simulate the model and control system in a sea state. The waves are introduced thru the rudder nondimensionalized coefficients as shown in chapter II.

The plots produced are shown in figures III-66 thru III-73. Figure III-65 was produced with the same program by setting $W_1=1.5$.

COMPUTER PROGRAM #7

```

//UHRINTF7 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
//EXEC DSL
//DSL.INPUT DD *
* TITLE RAS RUDDER CONTROL - SIMULATION WITH WAVE PERTURBATIONS
* RAS RUDDER CONTROL - SIMULATION WITH WAVE PERTURBATIONS
INTEGER NIS
CONST N=1, RD=1.0
PARAM IS=1, DD=0.2
PARAM RSENS=1.86642
PARAM WTSSENS=2.38692
PARAM RGN=23.41847
PARAM VFBC=4.35162
PARAM TSTP1=35.0, TSTP2=36.0
* HYDRODYNAMIC COEFFICIENTS
CONST MYVD=-0.00227, NV=-0.00351, NVD=-0.000197
CONST YV=-0.01243, YR=-0.0051, YNRD=0.00068, MXUD=0.0085
CONST YDEL=-0.0027, NDEL=-0.0012, YRD=-0.00027
CONST DLTD=2.0, DLTDM=7.0
* INITIAL SEPERATION
INCON X01=5.0, Y01=0.0, X02=0.0, Y02=0.4
* INITIAL CONDITIONS
INCON YAW01=0.0
INCON Y01=0.0, Y02=0.0, YN1=0.0, YN2=0.0
INCON U01=1.0, U02=1.5
PARAM WS=5.0
PARAM WD=-0.15.0
PARAM WL=1.0
PARAM WFMA=0.05685
INITIAL
DY0=Y02-Y01
DX0=X02-X01
CALL TRANS(YAW01, DX0, DY0, ADX, ADY)
CALL SLOPES(ADX, ADY, Y01, Y02, YN1, YN2)
CALCULATION OF THE COEFFICIENTS
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD

```

*


```

B21=MYR
A12=-NVD
B12=-NV
A22=IZNRD
B22=-NR
A33=MXUD
B33=-XU
KAI=-YDELR
KBI=NDELR
KCI=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
DRC=3.1415926/180.
LUC=20.84765
KG=DLTDM/DLTEM
DIDIC=0.0
D2=DEGRAD(1,1,D2D)
BY02=0.0
DERIVATIVE
WFI=(WFMA/(0.1137*40.0))*RAMP(0.0)
WF=LIMIT((-WFMA,WFMA,WFI)
WRV=NORMAL(1975,0.0,WFMA/10.0)
WV=WS/15.0
EWD=WD-YAWDP2
EWD=DEGRAD(1,1,EWDD)
WEF=2.0*3.1415926*(C DOT T2+WV*COS(EWD))/WL
WE=WEF*TIME
WY=WF*SIN(EWD)
WFN=WF*SIN(2.0*EWD)
WFX=WF*COS(EWD)
WNF=WFN*SIN(WE)+(3.1415926**WFY**2/WL)*SIN(2.0*WE)
WXF=WFX*SIN(WE)+(3.1415926**WFN**2/WL)*SIN(2.0*WE)
WY=WYF+WRV*WFY*SIN(WE)
WN=WNF+WRV*WFN*SIN(WE)
WX=WXF+WRV*WFX*SIN(WE)
DIDES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
DLTSI=LIMIT(-30.0,30.0,DIDES)
DLTEI=DLTSI-DID
DLTBEI=LIMIT(-DLTEM,DLTEM,DLTEI)
DID=INTGRL(DIDIC,KG*DLTBEI*LUC)
D1=DEGRAD(1,1,DID)
DX=X2-X1
DY=Y2-Y1
SIMULATION SHIP A
IF1=KAI*DI
IF21=KBI*DI

```

*


```

IF31=KC1*DI+NC1
I11=-B11*ADOT1-B21*BDOOT1+IF11
I21=-B12*ADOT1-B22*BDOOT1+IF21
I31=-B33*COOT1+IF31
ADDOOT1=(I11*A22-I21*A21)/D
BDDOOT1=(I21*A11-I11*A12)/D
CDDOOT1=I31/A33
ACOT1=INTGRL(O.,ADDOOT1)
BCOT1=INTGRL(O.,BDDOOT1)
CCOT1=INTGRL(UO1,CDDOOT1)
BY1=INTGRL(O.,BDOOT1)
XDOT1=COOT1*COS(BY1)-ADOT1*SIN(BY1)
YDOT1=COOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(XO1,XDOT1)
Y1=INTGRL(YO1,YDOT1)
YAW1=BY1
SWAY1=Y1
SURGE1=X1
SIMULATION SHIP B
IF12=KA1*D2+YY2+KA1*WY
IF22=KB1*D2+YN2+KB1*WN
IF32=KC1*D2+NC2+KC1*WX
I12=-B11*ADOT2-B21*BDOOT2+IF12
I22=-B12*ADOT2-B22*BDOOT2+IF22
I32=-B33*COOT2+IF32
ADDOOT2=(I12*A22-I22*A21)/D
BDDOOT2=(I22*A11-I12*A12)/D
CDDOOT2=I32/A33
ADOT2=INTGRL(O.,ADDOOT2)
BDOT2=INTGRL(O.,BDDOOT2)
CDOT2=SPDC TR(ADX,UO1,UO2)
BY2=INTGRL(BY02,BDOT2)
XDOT2=COOT2*COS(BY2)-ADOT2*SIN(BY2)
YDOT2=COOT2*SIN(BY2)+ADOT2*COS(BY2)
X2=INTGRL(XO2,XDOT2)
Y2=INTGRL(YO2,YDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2

```

NOSORT

```

YAWD1=DEGRAD(O,O,YAW1)
YAWDP1=DEGRAD(O,I,YAW1)
YAWD2=DEGRAD(O,O,YAW2)
YAWDP2=DEGRAD(O,I,YAW2)
YAWDPD=YAWDP1-YAWDP2
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RO,R1,B1,BB1,R2,B2,BB2)
CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,PSIADD,...)

```

*

*


```

PSIDED,WT,DA,AID,BID,B2D,WTSENS,DD,RD)
BDOT2D=DEGRAD(0,1,BDOT2)
BCOTFB=VFBG*BDOT2D
DDUMB=YAWD2-PSIDED+BDOTFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)

SORT
DISTE=ABS(DD-ADY)
CBJ=INTGRL(0.0,DISTE)

DYNAMIC REGION
* ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)
EXTERNAL FORCES ACTING BETWEEN SHIPS
CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
* ACTUAL TIME CONVERSION (SEC)
ATIME=LUC*TIME
AA1=(BB1+BB2)/2.0
CALL SWITCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D)

SAMPLE
CONTROL
PRINT
FINTIM=30.,DELT=0.04,DELS=0.04
0.20,X1,X2,DA,AID,PSIDFD,Y1,Y2,R1,BID,PSIADD,YAWD1,YAWD2,R2,...
B2D,PSIDED,ATIME,D2D,ADX,DID,YY2,BDOTFB,DLTS,ADY,DLTS1,...
YN2,CDOT1,CDOT2,OBJ,RSENS,VFBG,WYF,WNF,WXF,WRV,EWDD,WY,WN,...
WX,WE,WF

PRPLOT ONLY
CALL DRWG(1,1,ATIME,YAWDP2)
CALL DRWG(1,2,ATIME,YAWDP1)
CALL DRWG(2,1,ATIME,ADY)
CALL DRWG(3,1,ATIME,DLTS)
CALL DRWG(3,2,ATIME,D2D)
CALL DRWG(4,1,ATIME,YAWDPD)
CALL DRWG(5,1,ATIME,WX)
CALL DRWG(5,2,ATIME,WY)
CALL DRWG(5,3,ATIME,WN)

TERMINAL
IF(IS.EQ.1) WRITE(6,101)
101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')

```



```
CALL ENDRW(NPLOT)
CALL CONTIN
FINTIM=45.0
```

```
END
STOP
FORTRAN
INSERT SUBROUTINE SWITCH FROM APPENDIX A HERE
INSERT FUNCTION SPDCTR FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *
```

7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4

```
INSERT TWO /* CARDS HERE
```


COMPUTER PROGRAM #8

This program incorporates a fifth order polynomial curve fit speed control switching function to give optimal longitudinal positioning. The scenario is the same that was used in the design of the heading control development. The low order model of the gas turbine propulsion plant was used.

The plots produced are shown in figures III-80 thru III-83.

COMPUTER PROGRAM #8

```

//UHRINTF8 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
//EXEC DSL
//DSL.INPUT DD *
* TITLE RAS SPEED CONTROL - CONTROL TESTING
INTEGER NPLT=2
INTEGER NPLT=2
CONST N=1, RD=1.0
PARAM IS=1, DD=0.2
PARAM RSENS=1.86642
PARAM WTSENS=2.38692
PARAM WGN=23.41847
PARAM VFEG=4.35162
PARAM TSTP1=35.0, TSTP2=36.0
* HYDRODYNAMIC COEFFICIENTS
CONST NR=-0.00227, NV=-0.00351, NVD=-0.000197
CONST MYVD=0.015, MYR=0.0051, IZNRD=0.00068, MXUD=0.0085
CONST YV=-0.01243, XU=-0.0012, YRD=-0.00027
CONST YDELR=-0.0027, NDELR=-0.00126, XDELR=0.0
CONST DLITDM=2.0, DLITEM=7.0
* INITIAL SEPERATION
INCON X01=5.0, Y01=0.0, X02=0.0, Y02=0.4
* INITIAL CONDITIONS
INCON YAW01=0.0
INCON YY1=0.0, YY2=0.0, YN1=0.0, YN2=0.0
INCON U01=1.0, U02=1.5
PARAM UF=21.73, A=22.0, G=0.092
INITIAL
Y0=Y02-Y01
DX0=X02-X01
CALL TRANS(YAW01, DX0, DY0, ADX, ADY)
CALL SLOPES(ADX, ADY, YY1, YY2, YN1, YN2)
CALCULATION OF THE COEFFICIENTS
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=-MYR
A12=-NVD
B12=-NV

```



```

A22=IZNRD
B22=-NR
A33=MXUD
B33=-XU
KAI=-YDELR
KB1=NDELR
KC1=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
DRC=3.1415926/180.
LUC=20.84765
KG=DLTDM/DLTEM
DLDIC=0.0
C2=DEGRAD(1,1,D2D)
BY02=0.0
K=UF/A
P=4.88/LUC
SPDDER=U01/K
DERIVATIVE
DIDES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
DLTSL=LIMIT(-30.0,30.0,DIDES)
DLTSL=DLTSL-DID
DLTBE1=LIMIT(-DLTEM,DLTEM,DLTSL)
D1D=INTGRL(D1DIC,KG*DLTBE1*LUC)
D1=DEGRAD(1,1,D1D)
DX=X2-X1
DY=Y2-Y1
SIMULATION SHIP A
IF11=KA1*D1
IF21=KB1*D1+NC1
IF31=KC1*D1+NC1
I11=-B11*ADOT1-B21*BDO11+IF11
I21=-B12*ADOT1-B22*BDO11+IF21
I31=-B33*CDOT1+IF31
ADOT1=(I11*A22-I21*A21)/D
BDOT1=(I21*A11-I11*A12)/D
CDOT1=I31/A33
ACOT1=INTGRL(0.,ADOT1)
BCOT1=INTGRL(0.,BDOT1)
CCOT1=INTGRL(U01,CDOT1)
BY1=INTGRL(0.,BDOT1)
XDOT1=CDOT1*COS(BY1)-ADOT1*SIN(BY1)
YDOT1=CDOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(X01,XDOT1)
Y1=INTGRL(Y01,YDOT1)
YAW1=BY1
SWAY1=Y1

```

*


```

*
SURGE1=X1
SIMULATION SHIP B
IF12=KA1*Y2+YN2
IF22=KB1*Y2+YN2
IF32=KC1*Y2+YN2
IF12=-B11*ADOT2-B22*ADOT2-IF32
IF22=-B12*ADOT2-B22*ADOT2-IF32
IF32=-B13*ADOT2-B22*ADOT2-IF32
ADDOT2=(I12*A22-I12*A21)/D
BDOT2=(I12*A22-I12*A21)/D
CDDOT2=I32/A33
ACDOT2=INTGRL(0.,ADDOT2)
BCDOT2=INTGRL(0.,BDDOT2)
SW=SWCL(U01,U02)
SPDDEL=SPDRCL(ADX,U01,U02,SW)
SPDDEL=DELAY(7,P,SPDDEL,K-SPDDEL)
SPDEL=SPINIT(SPDEL,TIME,(SPDDEL/K-SPDDEL))
SPCIN=K*(SPDEL+SPDDEL)
SPDERR=(SPDIN-CDDOT2)*G
CDDOT2=INTGRL(U02,SPDERR*LUC)
BY2=INTGRL(BY02,BDOT2)
XDDOT2=CDDOT2*COS(BY2)-ADDOT2*SIN(BY2)
YDDOT2=CDDOT2*SIN(BY2)+ADDOT2*COS(BY2)
X2=INTGRL(X02,XDDOT2)
Y2=INTGRL(Y02,YDDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2
NOSORT
YAWD1=DEGRAD(0,0,YAW1)
YAWDPI1=DEGRAD(0,1,YAW1)
YAWD2=DEGRAD(0,0,YAW2)
YAWDP2=DEGRAD(0,1,YAW2)
YAWDPD=YAWDP1-YAWDP2
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
CALL PDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDF,PSIADD,...)
PSIDED=WT,DA,AD,B1D,B2D,WTSENS,DD,RD)
BCDOT2=DEGRAD(0,1,BDOT2)
BCOTFB=VFBG*BDOT2D
DDUMB=YAWD2-PSIDED+BDOTFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=DDUMB+360.0
DLTS=LIMIT(-30.0,30.0,DDUMB*RGN)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLT=M,DLTE,M,DLTBE*LUC)
D2D=INTGRL(D2DLC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)
*

```



```

SORT
DIST=ABS(DD-ADY)
OBJ=INTGRL(0.0,DISTE)
DISTES=ABS(ADX)
OBJJS=INTGRL(0.0,DISTES/25.0)
DYNAMIC REGION
* ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)
EXTERNAL FORCES ACTING BETWEEN SHIPS
CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
* ACTUAL TIME CONVERSION (SEC)
ATIME=LUC*TIME
AA1=(BB1+BB2)/2.0
CALL SWITCH(DD,DA,AA1,IS,RSENS,WTSSENS,RGN,VFBG,BDOT2D)
SAMPLE FINTIM=30.,DELT=0.04,DELS=0.04
CONTROL 0.20,ATIME,ADX,ADY,YAWD1,YAWD2,YAWDPD,SPDDES,CDOT2
PRINT PRPLOT ONLY
CALL DRWG(1,1,ATIME,ADX)
CALL DRWG(2,1,ATIME,SPDDES)
CALL DRWG(2,2,ATIME,CDOT2)
TERMINAL
IF (IS.EQ.1) WRITE(6,101)
101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
IF (IS.EQ.0) WRITE(6,102)
102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
CALL ENDRW(NPLOT)
CALL CONTIN
FINTIM=45.0
END
STOP
FORTRAN
INSERT SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT SUBROUTINE SWTCH FROM APPENDIX A HERE
INSERT FUNCTION SWCL FROM APPENDIX A HERE
INSERT FUNCTION SPDREC FROM APPENDIX A HERE
INSERT FUNCTION SPINIT FROM APPENDIX A HERE
INSERT SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN CD

```


7.0	5.0
7.0	5.0
7.0	5.0
7.0	5.0

4
4
4

INSERT TWO /* CARDS HERE

COMPUTER PROGRAM #9

This program introduces a longitudinal position offset capability. The method takes the control ship to the alongside position until 450 seconds into the run. After that time, with the ship steadied, the offset position desired (XCFSD) is switched to the desired offset. This method negates some of the transient oscillations which cause unstable conditions in the approach phase. A secondary change is the use of subroutine SWTCHF instead of subroutine SWTCH developed in the heading control section. This new subroutine relaxes the heading velocity feedback gain (VFEG) to allow turn stability in the turn phase.

The plots produced by this program are shown in figures III-84 thru III-101.

COMPUTER PROGRAM #9

```

//UHRINTF9 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=10
// EXEC DSL
//DSL INPUT DO *
* TITLE RAS SPEED CONTROL - OFFSET TESTING
* INTEG RKSFX
* INTEGER NPLOT
CONST NPLOT=2
* INTEGER N1 IS
CONST N1=1, RD=1.0
PARAM IS=1, DD=0.2
PARAM RSENS=1.86642
PARAM WSENS=2.38692
PARAM RGN=23.41847
PARAM VFBG=4.35162
PARAM TSTP1=35.0, TSTP2=36.0
* HYDRODYNAMIC COEFFICIENTS
CONST MYVD=0.00227, NV=-0.00351, NVD=-0.000197
CONST MYVD=0.015, MYR=0.0051, IZNRD=0.00068, MXUD=0.0085
CONST YV=-0.01243, XU=-0.0012, YRD=-0.00027
CONST YDELR=-0.0027, NDELR=-0.00126, XDELR=0.0
CONST DLTD=2.0, DLTDM=7.0
* INITIAL SEPERATION
INCON X01=5.0, Y01=0.0, X02=0.0, Y02=0.4
* INITIAL CONDITIONS
INCON YAW01=0.0, YN1=0.0, YN2=0.0
INCON U01=1.0, U02=1.5
* X0FS IS THE DESIRED NORMALIZED X POSITION / 0.0 IS ALONGSIDE /
* NEG FOR ASTERN OF ALONGSIDE / POS FOR FWD OF ALONGSIDE
* NOTE - X0FS CAN BE USED FOR VEHICLE OF APPROACH AND EXIT FROM
* STATION BY SETTING SOME DESIRED POSITION EG. -5.0, 5.0
* THIS WOULD CAUSE APPROACH OR BREAK AWAY STATION ATTAINMENT
* WHICH CAN BE PRECEDED BY OR FOLLOWED BY NON RAS POSITION
* CONTROL
PARAM XGFS=0.0
PARAM XCFSD=0.0
PARAM UF=21.73, A=22.0, G=0.092
INITIAL
DY0=Y02-Y01
DX0=X02-X01
CALL TRANS(YAW01, DX0, DY0, ADX, ADY)
CALL SLOPES(ADX, ADY, Y01, Y02, YN1, YN2)

```


*

CALCULATION OF THE COEFFICIENTS

NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=MYR
A12=-NVC
B12=-NV
A22=IZNRD
B22=-NR
A33=MXUD
B33=-XU
KA1=-YDELR
KB1=NDELR
KC1=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RCC=180./3.1415926
CRC=3.1415926/180.
LUC=20.84765
KG=CLTCM/DLTEM
DIDIC=0.0
D2=DEGRAD(1,1,D2D)
BY02=0.0
K=UF/A
P=4.88/LUC
SPC01=1.0
SPD02=1.5
SPDDER=SPD01/K

DERIVATIVE

DIDES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
DLTSL=LIMIT(-30.0,30.0,DIDES)
DLTBE1=DLTSL-DID
CLTBE1=LIMIT(-DLTEM,DLTEM,DLTBE1)
DIDC=INTGRL(DIDIC,KG*DLTBE1*LUC)
CI=DEGRAD(1,1,DID)
DX=X2-X1
DY=Y2-Y1
SIMULATION SHIP A

*

IF11=KA1*D1
IF21=KB1*D1
IF31=KC1*D1+NC1
I11=-B11*ADOT1-B21*BDOOT1+IF11
I21=-B12*ADOT1-B22*BDOOT1+IF21
I31=-B33*CDOT1+IF31
ACDOT1=(I11*A22-I21*A21)/D
BDCOT1=(I21*A11-I11*A12)/D


```

CCDOT1=I31/A33
ACDOT1=INTGRL(O.,ADDOT1)
BCDOT1=INTGRL(O.,BDDOT1)
CCDOT1=INTGRL(U01,CDDOT1)
BY1=INTGRL(O.,BDDOT1)
XDOT1=CDDOT1*COS(BY1)-ADOT1*SIN(BY1)
YDOT1=CDDOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(X01,XDOT1)
Y1=INTGRL(Y01,YDOT1)
YAW1=BY1
SWAY1=Y1
SURGE1=X1
SIMULATION SHIP B
IF22=KAL*D2+YY2
IF32=KB1*D2+YN2
I12=-B11*ADOT2-B21*BDDOT2+IF12
I32=-B12*ADOT2+IF32
ACDOT2=(I12*A22-I22*A21)/D
BDDOT2=(I12*A11-I12*A12)/D
CCDOT2=I32/A33
ADOT2=INTGRL(O.,ADDOT2)
BCDOT2=INTGRL(O.,BDDOT2)
SW=SWCL(SPDOT1,SPDOT2)
SPDDER=SPDDFC(ADX,SPDOT1,SPDOT2,SW,XOFS)
SPDELC=DELAY(7,P,(SPDDER/K-SPDDER))
SPDEL=SPINIT(SPDEL,TIME,(SPDDER/K-SPDDER))
SPDIN=K*(SPDEL+SPDDER)
SPDERR=(SPDIN-CDDOT2)*G
CCOT2=INTGRL(U02,SPDERR*LUC)
BY2=INTGRL(BY02,BDOT2)
XDOT2=CDDOT2*COS(BY2)-ADOT2*SIN(BY2)
YDOT2=CDDOT2*SIN(BY2)+ADOT2*COS(BY2)
X2=INTGRL(X02,XDOT2)
Y2=INTGRL(Y02,YDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2
NOSORT
YAWD1=DEGRAD(O.,YAW1)
YAWCP1=DEGRAD(O.,YAW1)
YAWD2=DEGRAD(O.,YAW2)
YAWCP2=DEGRAD(O.,YAW2)
YAWDPD=YAWDP1-YAWDP2
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,B1,BB1,R2,B2,BB2)
CALL HDGRAS(N,IS,R1,B1,BB1,R2,B2,BB2,PSIDFD,PSIADD,...

```

*

*


```

PSIDED,WT,DA,AID,B1D,B2D,WTSENS,DD,RD)
BCOT2D=DEGRAD(0,1,BDOT2)
BCOTFB=VF8G*BDOT2D
DCUMB=YAWD2-PSIDED+BDOTFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS=LIMIT(-30.0,30.0,DDUMB*RGD)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
D2C=INTGRL(D2DIC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)

SORT
DISTE=ABS(DD-ADY)
CBJ=INTGRL(0.0,DISTE)
DISTES=ABS(ADX)
OBJ=INTGRL(0.0,DISTES/25.0)

DYNAMIC REGION
* ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)
EXTERNAL FORCES ACTING BETWEEN SHIPS
CALL SLOPES(ADX,ADY,Y1,Y2,YN1,YN2)
IF(ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
100 FORMAT(' *****SEPARATION LESS THAN 25 FEET - COLLISION*****')
* ACTUAL TIME CONVERSION (SEC)
ATIME=LUC*TIME
AA1=(BB1+BB2)/2.0
CALL SWITCH(DD,DA,AA1,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
IF (ATIME.GT.450.0) XOFS=XOFSD

SAMPLE FINIM=30.,DELT=0.04,DELS=0.04
CONTRL 0.20,ATIME,ADX,ADY,YAWD1,YAWD2,YAWDPD,SPDDDES,CDOT2
PRPLOT ONLY
CALL DRWG(1,1,ATIME,ADY)
CALL DRWG(2,1,ATIME,YAWDPD)
CALL DRWG(3,1,ATIME,SPDDDES)
CALL DRWG(3,2,ATIME,CDOT2)
CALL DRWG(4,1,ATIME,ADX)

TERMINAL
IF(IS.EQ.1) WRITE(6,101)
101 FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
102 FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
CALL ENDRW(NPLOT)
CALL CONTIN
FINIM=45.0
END

```



```

STOP
FORTRAN
SUBROUTINE RBMEAS FROM APPENDIX A HERE
SUBROUTINE HDGRAS FROM APPENDIX A HERE
FUNCTION DEGRAD FROM APPENDIX A HERE
SUBROUTINE TRANS FROM APPENDIX A HERE
SUBROUTINE SWTCHF FROM APPENDIX A HERE
FUNCTION SWCL FROM APPENDIX A HERE
FUNCTION SPDOFC FROM APPENDIX A HERE
SUBROUTINE SPINIT FROM APPENDIX A HERE
SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *

```

7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4
7.0	5.0	4

INSERT TWO /* CARDS HERE

COMPUTER PROGRAM #10

This program incorporates the sea state first programmed in computer programs #4 and #7. The WX wave force, however, is introduced at the end of the propulsion loop to allow more realistic perturbations. This is the final form of the complete heading and speed control systems. To run this without a sea state, set WFMA to 0.0.

This program produced the plots shown in figures III-103 thru III-105.


```
//UHRINTF0 JOB (2794,0775,EA44),'UHRIN SMC 1675',TIME=5
```

363


```

INITIAL
DYO=Y02-X01
DX0=X02-X01
CALL TRANS(YAW01,DX0,DYO,ADX,ADY)
CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
CALCULATION OF THE COEFFICIENTS
NC1=-XU
NC2=-XU
A11=MYVD
B11=-YV
A21=-YRD
B21=MYR
A12=-NVD
B12=-NV
A22=IZNRD
B22=-NR
A33=MXUD
B33=-XU
KAI=-YDELR
KBI=XDELR
KCI=XDELR
D=A11*A22-A12*A21
DELRM=41.6953
RDC=180./3.1415926
LUC=20.84765
KG=DLTDM/DLTEM
D1DIC=0.0
D2=DEGRAD(1,1,D2D)
BY02=0.0
K=UF/A
P=4.88/LUC
SPD01=1.0
SPD02=1.5
SPDDER=SPD01/K
DERIVATIVE
WFI=(WFMA/(0.1137*40.0))*RAMP(0.0)
WFLIMIT=(-WFMA,WFMA,WFI)
WRV=NORMAL(1975,0.0,WFMA/10.0)
WV=WS/15.0
EWCD=WD-YAWDP2
EWD=DEGRAD(1,1,EWDD)
WEF=2.0*3.1415926*(CDOT2+WV*COS(EWD))/WL
WE=WEF*TIME
WFI=WF*SIN(EWD)
WFN=WF*SIN(2.0*EWD)
WFX=WF*COS(EWD)
WYF=WFY*SIN(WE)+(3.1415926*WFY**2/WL)*SIN(2.0*WE)

```



```

WNF=WFN*(WE)+(3.1415926**WFN**2/WL)*SIN(2.0**WE)
WXF=WFN*(WE)+(3.1415926**WFN**2/WL)*SIN(2.0**WE)
WY=WFN*(WE)+(3.1415926**WFN**2/WL)*SIN(2.0**WE)
WN=WFN*(WE)+(3.1415926**WFN**2/WL)*SIN(2.0**WE)
WX=WFN*(WE)+(3.1415926**WFN**2/WL)*SIN(2.0**WE)
DIDES=5.0*STEP(TSTP1)-5.0*STEP(TSTP2)
DLTS1=LIMIT(-30.0,30.0,DIDES)
DLTE1=DLTS1-DID
DLTBE1=LIMIT(-DLTEM,DLTEM,DLTE1)
DID=INTGRL(DIDIC,KG*DLTBE1*LUC)
DI=DEGRAD(1,1,DID)
DX=X2-X1
DY=Y2-Y1
SIMULATI=ON SHIP A
IF11=KAI*DI
IF21=KBI*DI
IF31=KCI*ABS(DI)+NC1
I11=-B11*ADOT1-B21*BDDOT1+IF11
I21=-B12*ADOT1-B22*BDDOT1+IF21
I31=-B13*ADOT1-B23*BDDOT1+IF31
ACDOT1=(I11*A22-I21*A21)/D
BCDOT1=(I21*A11-I11*A12)/D
CDDOT1=I31/A33
ADCT1=INTGRL(0.,ADDOT1)
BCOT1=INTGRL(0.,BDDOT1)
COT1=INTGRL(U01,CDDOT1)
BY1=INTGRL(0.,BDDOT1)
XDOT1=CDDOT1*COS(BY1)-ADOT1*SIN(BY1)
YDOT1=CDDOT1*SIN(BY1)+ADOT1*COS(BY1)
X1=INTGRL(X01,XDOT1)
Y1=INTGRL(Y01,YDOT1)
YAW1=BY1
SWAY1=Y1
SURGE1=X1
SIMULATI=ON SHIP B
IF12=KAI*D2+YY2+KAI*WY
IF22=KBI*D2+YN2+KBI*WN
IF32=KCI*ABS(D2)+NC2
I12=-B11*ADOT2-B21*BDDOT2+IF12
I22=-B12*ADOT2-B22*BDDOT2+IF22
ADDOT2=(I12*A22-I22*A21)/D
BDDOT2=(I22*A11-I12*A12)/D
ADOT2=INTGRL(0.,ADDOT2)
BCOT2=INTGRL(0.,BDDOT2)
SW=SWCL(SPD01,SPD02)
SPDDES=SPD0FC(ADX,SPD01,SPD02,SW,XOFS)
SPDDEL=DELAY(7,P,(SPDDES/K-SPDDES))
SPDEL=SPINIT(SPDDEL,TIME,(SPDDES/K-SPDDES))

```

*

*


```

SPDIN=K*(SPDEL+SPDDER)
SPDERR=(SPDIN-CDOT2)*G
CDOT2 = INTGRL(U02,SPDERR*LUC)+KS2*WX
BY2=INTGRL(BY02,BDOT2)
XDOT2=CDOT2*COS(BY2)-ADOT2*SIN(BY2)
YDOT2=CDOT2*SIN(BY2)+ADOT2*COS(BY2)
X2=INTGRL(X02,XDOT2)
Y2=INTGRL(Y02,YDOT2)
YAW2=BY2
SWAY2=Y2
SURGE2=X2

NOSORT
YAWD1=DEGRAD(0,0,YAW1)
YAWDP1=DEGRAD(0,1,YAW1)
YAWD2=DEGRAD(0,0,YAW2)
YAWDP2=DEGRAD(0,1,YAW2)
YAWDPD=YAWDP1-YAWDP2
RUDDER RESPONSE INPUT
CALL RBMEAS(N,YAW1,X1,Y1,YAW2,X2,Y2,RD,R1,BB1,R2,B2,BB2)
CALL HDGRAS(N,IS,R1,BB1,R2,B2,BB2,RSENS,YAW2,PSIDFD,PSIADD,...
PSIDED,WT,DA,AID,BID,B2D,WTSENS,DD,RD)
BDOT2D=DEGRAD(0,1,BDOT2)
BDOTFB=VFBG*BDOT2D
DDUMB=YAWD2-PSIDED+BDOTFB
IF(DDUMB.GT.180.0) DDUMB=DDUMB-360.0
IF(DDUMB.LT.-180.0) DDUMB=360.0+DDUMB
DLTS=LIMIT(-30.0,30.0,DDUMB*RGD)
DLTE=DLTS-D2D
DLTBE=LIMIT(-DLTEM,DLTEM,DLTE)
D2D=INTGRL(D2DIC,KG*DLTBE*LUC)
D2=DEGRAD(1,1,D2D)

SORT
DISTE=ABS(DD-ADY)
OBJ=INTGRL(0.0,DISTE)
DISTES=ABS(ADX)
OBS=INTGRL(0.0,DISTES/25.0)

DYNAMIC REGION ACTUAL SEPARATION
DX=X2-X1
DY=Y2-Y1
CALL TRANS(YAW1,DX,DY,ADX,ADY)
EXTERNAL FORCES ACTING BETWEEN SHIPS
CALL SLOPES(ADX,ADY,YY1,YY2,YN1,YN2)
IF((ABS(ADY).LT.0.04744).AND.(ABS(ADX).LT.1.0)) WRITE(6,100)
FORMAT(1,***SEPARATION LESS THAN 25 FEET - COLLISION****)
100 ACTUAL TIME CONVERSION (SEC)
ACTUAL TIME*TIME
ATIME=LUC*TIME
AA1=(BB1+BB2)/2.0

```



```

CALL SWTCHF(DD,DA,AAL,IS,RSENS,WTSENS,RGN,VFBG,BDOT2D,XOFS)
IF (ATIME.GT.450.0) XOFS=XOFS0

SAMPLE  FINTIM=30.,DELT=0.04,DELS=0.04
CONTRL  0.20,ATIME,YAWDI,DID,DLTS1,CDO1,OBJ,YAWD2,D2D,DLTS,CDO2,...
PRINT   OBJ,YAWDPD,ADY,ADX,SPDDDES,EMDD,WEF,WY,WN,WX,RSENS,WTSENS,...
PRPLOT  RGN,VFBG,SW

ONLY    CALL DRWG(1,1,ATIME,CDO2)
CALL DRWG(1,2,ATIME,SPDDDES)
CALL DRWG(2,1,ATIME,ADX)

TERMINAL
IF (IS.EQ.1) WRITE(6,101)
101  FORMAT(' THIS RUN IS FOR A PORT SIDE TO APPROACH')
102  IF (IS.EQ.0) WRITE(6,102)
    FORMAT(' THIS RUN IS FOR A STBD SIDE TO APPROACH')
CALL ENDRW(NPLOT)
CALL CONTIN
FINTIM=45.0

END
STOP
FORTRAN
INSERT  SUBROUTINE RBMEAS FROM APPENDIX A HERE
INSERT  SUBROUTINE HDGRAS FROM APPENDIX A HERE
INSERT  FUNCTION DEGRAD FROM APPENDIX A HERE
INSERT  SUBROUTINE TRANS FROM APPENDIX A HERE
INSERT  SUBROUTINE SWTCHF FROM APPENDIX A HERE
INSERT  FUNCTION SWCL FROM APPENDIX A HERE
INSERT  FUNCTION SPDOFC FROM APPENDIX A HERE
INSERT  FUNCTION SPINIT FROM APPENDIX A HERE
INSERT  SUBROUTINE SLOPES FROM APPENDIX A HERE
//PLOT.SYSIN DD *

INSERT TWO /* CARDS HERE

```

4
4
4
4

7.0 5.0
7.0 5.0
7.0 5.0
7.0 5.0

BIBLIOGRAPHY

1. Calvanc, C. N., An Investigation of the Stability of a System of Two Ships Employing Automatic Control while on Parallel Courses in Close Proximity, M.S. Thesis, Massachusetts Institute of Technology, Cambridge, 1970.
2. Principles of Naval Architecture, revised ed., p.466-470, The Society of Naval Architects and Marine Engineers, 1967.
3. Principles of Naval Architecture, revised ed., p.477-486, The Society of Naval Architects and Marine Engineers, 1967.
4. Naval Ship Research and Development Center Report MEL 202/67, Development of a Simulation Model of the Steering-Control System for a Destroyer Class Ship, by C.L. Patterson Jr., March 1969.
5. Van, T. V., Simulation of Propulsion Plant Dynamics and Their Effect on Speed Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.
6. Naval Ship Research and Development Center Report 27-745, Simulation of Maneuvering Characteristics of a Destroyer Study Ship Using a Modified Nonlinear Model, by S. H. Ercwin and R. Alvestad, August 1974.
7. Thaler, G. J., "Computer Determination of Low Order Models for High Order Systems," Computers and Electrical Engineering, v. 2, p. 117-123, 1975.
8. Massachusetts Institute of Technology Department of Naval Architecture and Marine Engineering Report no. 66-5, The Analysis and Modeling of Irregular Waves, by F. B. Sellers and T. A. Loukakis, July 1966.
9. Naval Ship Research and Development Center Report 6-106/70, Digital Computer Simulation of Sea Wave Height / Time Series, by A. S. Fields, December 1970.
10. Lima, C. G., Multivariable Systems Design: A Two Ship Controller for Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.
11. Astorquiza Vivar, G. M., Automatic Control System for Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1975.
12. Sarzetakis, T., Maneuvering Control of Replenishment at Sea, M.S. Thesis, Naval Postgraduate School, Monterey, 1972.
13. Aguayo, F., Course Keeping with Automatic Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1973.
14. Mather, T. S., Precision Surface Ship Separation Measurement and Maneuvering Control Utilizing Radar Techniques, paper presented at Ship Control Systems Symposium, 2nd, Annapolis, Maryland, 4-5-6 November 1969.
15. Aseltine, J. A., Mancini, A. R., and Sarture, C. W., "A

BIBLIOGRAPHY (cont.)

Survey of Adaptive Control Systems," IRE Transactions on Automatic Control, v. PGAC-6, p. 102-108, December 1958.

16. Leonides, C. T., Modern Control Systems Theory, McGraw-Hill, 1965.
17. National Aeronautics and Space Administration NASA CR-669, Theoretical and Experimental Research on Digital Adaptive Control System, by J. Zaborszky, R. G. Marsh, E. E. Janitch, M. R. Chidambara, and E. E. Buder, January 1967.
18. National Aeronautics and Space Administration NASA CR-810, Continuation of Theoretical and Experimental Research on Digital Adaptive Control Systems, by J. Zaborszky, R. G. Marsh, R. E. Janitch and M. R. Chidambara, July 1976.
19. Naval Electronics Laboratory NEL Report 1216, Adaptive Plant Identification Digital Control System, by W. V. Kershaw, 30 March 1964.
20. Defense Documentation Center TR No. ASD-TR-61-27 Volume I, Fundamental Study of Adaptive Control Systems, by R. E. Kalman, T. S. Englar, and R. S. Bucy, April 1962.
21. National Aeronautics and Space Administration NASA CR-715, Study of Optimal and Adaptive Control Theory, by C. D. Johnson, April 1967.
22. Mishkin, E. and Braun Jr., L., Adaptive Control Systems, McGraw-Hill, 1961.
23. Cadzow, J. A., Martens, H. R., Discrete-Time and Computer Control Systems, Prentice-Hall, Inc., 1973.
24. Aguayo, E., Course-Keeping with Automatic Control, M.S. Thesis, Naval Postgraduate School, Monterey, 1973.
25. Hozos, A. G., Maneuvering Characteristics of Automatically Controlled Ships with Directionally Unstable Hulls, M.S. Thesis, Naval Postgraduate School, Monterey, 1974.
26. Navy Marine Engineering Laboratory MEL Report 333/65, Evaluation of Functional Performance of an Integrated Ship Control Conning Console by Operator Personnel, by J. I. McIane, W. J. Weingartner, and J. C. Townsend, May 1966.
27. Hess, D., "Shipboard Manning Reduction," All Hands, Nr. 707, p. 14-21, December 1975.
28. "Navy Commands Report 4 More Ship Collisions," Navy Times, p. 4, 31 December 1975.
29. Naval Research and Development Center Report 76-0040, Automatic Control of Underway Replenishment Maneuvers in Handcru Seas, by R. Alvestad, April 1976.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4. Professor George J. Thaler, Code 52Tr Naval Postgraduate School Monterey, California 93940	5
5. Lt. John J. Uhrin III, USN 456 Lineberry Road Virginia Beach, Virginia 23452	3
6. Reidar Alvestad NSREC Annapolis Lab Annapolis, Maryland 21402	1
7. Assoc. Professor Alex Gerba, Jr., Code 52Gz Naval Postgraduate School Monterey, California 93940	2
8. Samuel E. Brown NSREC Annapolis Lab Annapolis, Maryland 21402	1

INITIAL DISTRIBUTION LIST (cont.)

No. Copies

- | | | |
|-----|--|---|
| 9. | J. K. Zuidweg | 1 |
| | Royal Netherlands Naval College | |
| | Den Helder, The Netherlands | |
| 10. | RADM R. L. Walters | 1 |
| | Project Manager Surface Ship Project Office PM18 | |
| | Room 9SC8 National Center 3 | |
| | Department of the Navy | |
| | Washington, DC 20362 | |
| | Attention PM18T | |
| 11. | Marvin Denicoff | 1 |
| | Office of Naval Research Code 437 | |
| | Arlington, Virginia 22217 | |



Thesis

165719

U15 Uhrin

c.1 Sampled data adaptive
digital computer control
of surface ship maneuvers.

1 JUN 79
27 JUL 89

25585
35218

Thesis

165719

U15 Uhrin

c.1 Sampled data adaptive
digital computer control
of surface ship maneuvers.

thesU15

Sampled data adaptive digital computer c



3 2768 001 11247 7

DUDLEY KNOX LIBRARY